



Étudiant : David Rapin
Suiveur UTC : Djamal Boukerroui



UTC : GI, Filière ICSI
Semestre : Printemps 2007

Conception, développement et mise en œuvre d'un système de collecte de vidéos en ligne

Entreprise : Institut National de l'Audiovisuel
Lieu : Bry-sur-Marne
Responsable : Thomas Drugeon

Remerciements

Je souhaite remercier les personnes qui ont rendu possible ce stage, ou qui ont contribué à son bon déroulement, en particulier:

Bruno Bachimont, responsable scientifique à l'INA, pour m'avoir permis de trouver ce stage;

Daniel Teruggi, Directeur de la recherche à l'INA, pour m'avoir accueilli au sein de la Direction Recherche et Expérimentation;

Thomas Drugeon, responsable du projet du Dépôt Légal du Web, pour son soutien, sa patience et son aide;

Frédéric Dumas, responsable du projet Signature, pour sa disponibilité et son écoute;

Djamal Boukerroui, enseignant chercheur à l'UTC, pour m'avoir suivi pendant ce stage.

Je remercie enfin tous ceux qui ont été présents à la D.R.E. durant mon stage et qui ont contribué à en faire une expérience agréable au quotidien: Jérôme, Hervé, Marie-Luce, Pierre, Léguman, Ludovic, Quentin, Jeremy, Jean-Pascal.

Sommaire

I. Résumé Technique.....	3
II. Présentation du cadre du projet.....	4
1. L'institut National de l'Audiovisuel.....	4
a. Statut et missions.....	4
b. Moyens.....	5
c. Deux visages : conservation du patrimoine et recherche.....	6
2. Le Dépôt Légal.....	8
a. Origine du Dépôt Légal.....	8
b. Organismes concernés par le Dépôt Légal.....	9
3. Le projet du Dépôt Légal du Web à l'INA.....	10
a. Motivations et cadre.....	10
b. Objectifs et caractéristiques.....	11
c. Description technique globale.....	12
III. Projet de fin d'études.....	13
1. Objectifs.....	13
2. Rôles occupés dans les projets.....	14
3. Antécédents nécessaires.....	15
4. Plannings du travail.....	16
5. Librairie de capation en Perl : VideoScrap.....	17
a. Compréhension du framework et des outils de développement.....	17
b. Présentation des procédés effectifs d'intégration de vidéos.....	19
c. Architecture et implémentation de la librairie.....	23
d. Interface de monitoring et de contrôle.....	26
e. Mise en oeuvre de la librairie.....	29
6. Extension Firefox : GetVideo.....	31
a. Objectifs et besoins spécifiques.....	31
b. Étude de l'existant.....	32
c. Principes de développement d'extensions pour Firefox.....	33
d. Principe et implémentation de l'extension.....	34
e. Utilisation de l'extension Firefox.....	36
IV. Synthèse.....	38
1. Projet de librairie Perl.....	38
2. Projet d'extension Firefox.....	39
3. Conclusion.....	40
V. Bibliographie.....	41
VI. Annexes.....	42
1. Internet a-t-il une mémoire ?	42
2. La mémoire du flux : Entretien avec Emmanuel Hoog.....	43
3. Cadre légal du Dépôt Légal du Web : la DADVSI.....	46
4. Diagramme de classes simplifié de la librairie VideoScrap.....	48
5. Collecte du traitement Web de la campagne présidentielle 2007.....	49

I. Résumé Technique

Mon stage a consisté en la résolution du problème du téléchargement de vidéos intégrées dans des pages Web, dans le cadre de deux projets de recherche distincts à l'INA.

La partie principale de mon stage a été le développement d'une librairie de programmes en langage Perl, destinée à être utilisée avec les outils de collecte du projet du Dépôt Légal du Web (DLWeb) pour la captation des vidéos contenues dans les pages Web archivées.

Une seconde partie de mon stage a été consacrée au développement d'une extension du navigateur Mozilla Firefox, en XUL¹ et JavaScript, destinée à permettre à des utilisateurs non-expérimentés de télécharger sur leur disque dur les vidéos contenues dans des pages Web. Cette partie a été réalisée dans le cadre du projet Signature.

Le développement Perl et Firefox s'est principalement effectué sur un serveur distant Linux Fedora. Le développement sur ce serveur était rendu possible par le partage des fichiers de code via Samba². La gestion des versions et des modifications, assurant la cohérence du code modifié par plusieurs personnes à la fois, était effectuée à l'aide d'un serveur SVN³. L'accès au code et le travail quotidien s'est fait sur un poste PC sous Windows XP, dont mes outils principaux étaient un navigateur Web (Mozilla Firefox) et un éditeur de texte spécialisé dans la programmation (SciTe).

Outre l'aspect purement *programmation* de mon stage, l'univers audiovisuel a représenté pour moi un ensemble de pratiques, formats et standards à connaître et à comprendre. La connaissance des différents protocoles de consultation (HTTP, MMS, RTSP), des différents formats vidéos (FLV, MPEG, WMV, RM) et des spécificités des serveurs proposant ces vidéos a requis l'assimilation de nombreuses techniques et d'un savoir-faire spécifique.

1 XML-based User interface Language : langage de description d'interfaces graphiques basé sur XML, créé dans le cadre du projet Mozilla.

2 Samba est un logiciel libre supportant le protocole SMB/CIFS. Ce protocole est employé par Microsoft pour le partage de diverses ressources (fichiers, imprimantes, etc.) entre ordinateurs équipés de Windows. Samba permet aux systèmes Unix d'accéder aux ressources de ces systèmes et vice-versa.

3 Subversion (en abrégé SVN) est un logiciel libre dédié à la gestion de versions, et s'appuyant entre autres sur le principe du dépôt centralisé et unique ainsi que sur la propagation atomique des modifications.

II. Présentation du cadre du projet

1. L'institut National de l'Audiovisuel

a. Statut et missions

L'Institut National de l'Audiovisuel est un établissement public de l'Etat à caractère Industriel et Commercial (EPIC), né de l'éclatement de l'ORTF⁴ en 1974. L'INA a pour vocation d'archiver et d'indexer des documents télévisuels et radiophoniques.

Détenteur d'une des collections audiovisuelles les plus importantes au monde, avec celles de la BBC⁵ et de la RAI⁶, ses archives réunissent plus d'un million et demi d'heures de radio et de télévision, soit deux millions et demi de documents audiovisuels répartis sur quatre-vingts kilomètres de rayonnages.

Les fonds d'archives de l'INA constituent une source documentaire majeure pour les professionnels de l'audiovisuel et du multimédia, ainsi que pour les chercheurs, enseignants et étudiants. Tous les genres de la télévision et de la radio sont représentés [1].

Ses principales activités s'articulent autour de trois thèmes :

- **La conservation du patrimoine audiovisuel national :**
 - Assurer la collecte des programmes audiovisuels
 - Préserver et restaurer les fonds
 - Offrir des services documentaires renouvelés et efficaces
- **L'exploitation et la mise à disposition du patrimoine :**
 - Développer l'exploitation commerciale des fonds
 - Valoriser les archives à des fins scientifiques, éducatives et culturelles
- **L'accompagnement des évolutions du secteur audiovisuel à travers ses activités de recherche, de production et de formation :**
 - Renforcer la convergence des activités de recherche et d'expérimentation vers la mission patrimoniale
 - Accroître le caractère innovant de la production de création et de recherche
 - Orienter la formation professionnelle vers les technologies numériques

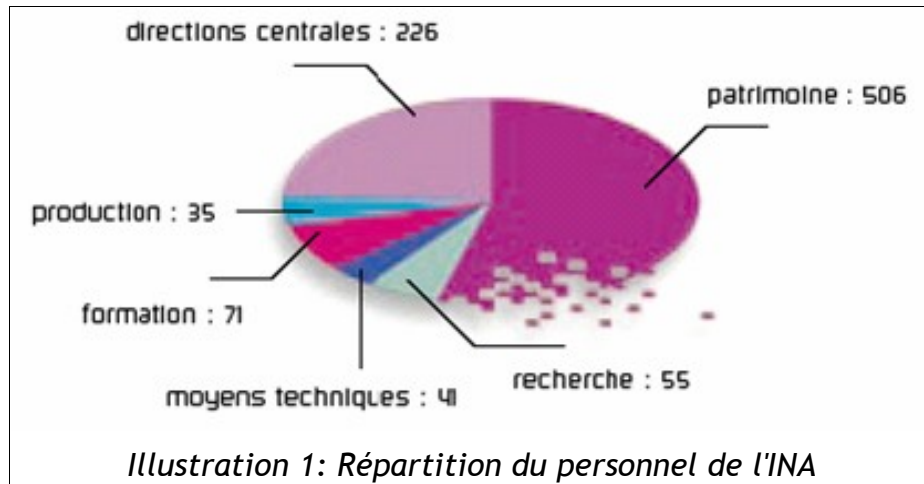
4 Office de Radiodiffusion - Télévision Française

5 British Broadcasting Corporation

6 Radio Audizioni Italiane : service public de radio-télévision en Italie

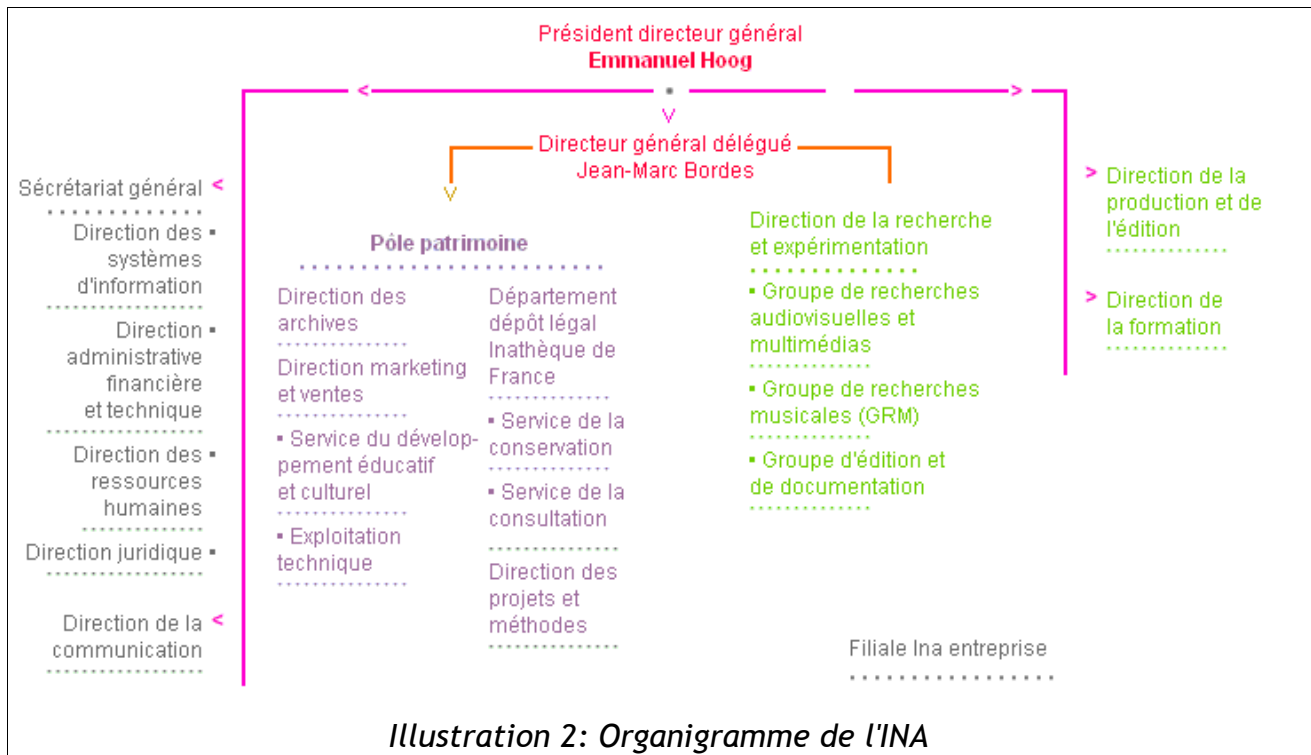
b.Moyens

La masse salariale de l'INA en 2006 s'élevait à 955 salariés (équivalents temps plein) répartis comme suit :



En tant que EPIC, l'INA est financé à hauteur de 60% par les redevances publiques et de 40% par les recettes contractuelles. Le dernier rapport d'activité (2006) indique un chiffre d'affaires de 107.468 M€, avec un résultat net de 1.7 M€.

c. Deux visages : conservation du patrimoine et recherche



Le pôle patrimoine, qui représente la mission principale de l'INA, est composé de deux départements : le Département Droits et Archives (DDA) et l'Inathèque de France.

- Le DDA a pour mission l'archivage professionnel **pour les chaînes publiques de radio-télévision**. Il est ainsi responsable de la gestion de 500 000 heures de télévision, 600 000 heures de radio et plus d'un million de photos
- L'Inathèque de France a été créée le 1er janvier 1995. Il est chargé de la gestion du dépôt légal de la radio-télévision défini par la loi du 20 juin 1992. Le décret d'application de cette loi énoncé le 31 décembre 1993 **concerne les diffuseurs nationaux hertziens**, les émissions d'origine française en première diffusion et les documents écrits d'accompagnement (conducteur d'émission, rapport de chef de chaîne, dossier de presse...). Cette loi conduit à la conservation annuelle moyenne de 500 000 heures de TV (sur 50 000 diffusées) pour 80 chaînes, 17 500 heures de radio (sur 40 000 diffusées) pour 5 chaînes nationales, 35 000 documents écrits pour 12 chaînes de radio-télévision (50 000 pages). Une partie de ces archives est accessible au public sur le site de l'INA⁷ depuis 2006.

Les activités de recherche de l'INA s'effectuent sous la Direction de la Recherche et Expérimentation (DRE), composée du Groupe de Recherches Audiovisuelles et Multimédia (GRAM) et du Groupe de Recherche Musicales (GRM). Les activités s'articulent autour des thèmes suivants :

- Description des contenus audiovisuels (DCA)
- Traitement technique de l'audiovisuel
- Visualisation, interfaces, expérimentation (VIE)
- Interprétation sémiotique de l'audiovisuel
- Étude socio-économique des médias

⁷ <http://www.ina.fr>

La recherche audiovisuelle s'oriente ainsi principalement sur la mise au point d'outils numériques dans le domaine de la restauration et de l'indexation de documents audiovisuels.

L'INA dispose également de secteurs opérationnels qui sont :

- La DCPR (Direction de Programmes de la Création et de la Recherche) qui conçoit et réalise des programmes audiovisuels, et hypermédia.
- INA-Formation : le plus grand centre européen de formation professionnelle aux métiers de l'audiovisuel.
- Le Service de Moyens Techniques, qui met à disposition des deux autres secteurs opérationnels un ensemble de moyens techniques et humains.

2. Le Dépôt Légal

a. Origine du Dépôt Légal

Instauré par François Ier en 1537 pour les oeuvres écrites, puis étendu à d'autres supports, le Dépôt Légal est organisé en vue de permettre :

- La collecte et la conservation des documents de toute nature publiés, produits ou diffusés en France
- La constitution et la diffusion de bibliographies nationales.
- La consultation des documents, sous réserve des secrets protégés par la loi, dans des conditions conformes à législation sur la propriété intellectuelle et compatible avec leur conservation.

Le Dépôt Légal s'applique aujourd'hui aux documents imprimés, graphiques, photographiques, sonores, audiovisuels, multimédias, quel que soit leur procédé technique de fabrication, d'édition ou de diffusion, dès lors qu'ils sont mis à la disposition du public. Il concerne également les logiciels, les bases de données, les systèmes experts et les autres produits de l'intelligence artificielle dès lors qu'ils sont mis à la disposition du public par la diffusion d'un support matériel, quelle que soit la nature de ce support.

b. Organismes concernés par le Dépôt Légal

La recherche d'une meilleure adéquation entre le Dépôt Légal et le champ culturel contemporain conduit à élargir la liste des documents concernés par l'obligation de Dépôt et à répartir la responsabilité de leur gestion entre plusieurs organismes spécialisés dans leurs domaines respectifs [2]:

- La Bibliothèque Nationale de France (BnF) est chargée d'archiver les documents imprimés et graphiques de toutes sortes, notamment les livres, périodiques, brochures, estampes, gravures, cartes postales, affiches, cartes, plans, globes et atlas géographiques, partitions musicales, chorégraphies et documents photographiques, ainsi que les progiciels, bases de données et systèmes experts, les phonogrammes de toutes natures, les vidéogrammes non fixés sur support photochimique, les documents multimédias.
- L'institut National de l'Audiovisuel est responsable des documents sonores et audiovisuels radiodiffusés ou télédiffusés.
- Le Centre National de la cinématographie gère les films sur support photochimique. Cette définition va évoluer avec l'arrivée inéluctable de la diffusion en numérique dans les salles de projection.
- Le Ministère de l'intérieur est responsable des archives concernant les livres, brochures et documents imprimés de toute nature édités ou importés sur le territoire métropolitain ainsi que les périodiques édités ou importés dans les départements métropolitains ainsi que les périodiques édités ou importés dans les départements métropolitains et d'outre-mer.

3. Le projet du Dépôt Légal du Web à l'INA

a. Motivations et cadre

Depuis sa création, il y a plus de 400 ans, le Dépôt Légal s'est adapté aux nouveaux médias à mesure que leur intérêt patrimonial devenait évident. Le Web représente aujourd'hui un nouveau support de communication diffusant des informations et des contenus dont la valeur patrimoniale ne peut être ignorée⁸.

Les législateurs s'intéressent ainsi à définir le cadre juridique du Dépôt Légal des sites Web français. La charge de ce Dépôt Légal sera partagée entre l'INA et la BnF⁹, dans la continuité des collections des deux institutions. Fort de son savoir-faire dans le domaine du Dépôt Légal Radio/Télévision, l'INA se concentrera ainsi essentiellement sur les sites relevant des industries culturelles et des médias radios et télévisions. La BnF quant à elle s'intéressera aux publications et collections en ligne, ce qui représente une quantité plus importante de documents mais un volume équivalent à celui attaché à l'INA.

Cependant le Web est un média complexe à la fois spatialement mais également temporellement en raison de son évolution permanente. On le définit comme un réseau hypermédia ouvert. L'évolution de ses contenus ou de sa structure n'est pas gérée par une autorité homogène. La constitution de l'archive du Web doit tenir compte de ces propriétés encore inédites pour un espace documentaire. Dans un premier temps, l'enjeu est de produire des copies conformes d'une sous partie du Web à une fréquence relativement élevée.

Cependant, connaître cette sous-partie du Web n'est pas chose aisée face à l'organisation complexe et à la quantité des documents. Même en réduisant cet espace à un seul thème et à un instant t , la quantité et le volume des documents présents sur Web désorientent les utilisateurs (experts ou occasionnels) voulant les explorer. En parallèle de la constitution d'une telle mémoire, il apparaît nécessaire de développer de nouveaux concepts afin de donner du sens à cet espace vaste, non organisé et évoluant dans le temps.

Ainsi, l'INA s'intéresse également à rendre intelligible cette mémoire du réseau en étudiant des dispositifs de lecture du Web permettant de recontextualiser l'information dans le temps et dans l'espace. Des outils de visualisation radicalement différents de ceux utilisés dans la radio-télévision, seront également à développer pour naviguer dans cette archive.

C'est dans ce cadre que l'INA mène des études et des expérimentations sur les techniques, procédés et méthodes permettant la captation des contenus, l'organisation de la mémoire et la constitution d'une archive du Web depuis 2000.

Le cadre légal du Dépôt Légal du Web a été fixé par le Titre IV de la loi relative au droit d'auteur et aux droits voisins dans la société de l'information (DADVSI)¹⁰. Néanmoins, le débat qui a entouré le vote de cette loi, concernant des articles jugés liberticides, a retardé sa mise en application en raison de l'échéance électorale de 2007.

Par conséquent, en l'absence du décret d'application de cette loi, les expérimentations de préfiguration sont encore menées sans budget supplémentaire ni cadre définitif.

⁸ voir à ce sujet: Annexe 1 - Internet a-t-il une mémoire ?

⁹ voir à ce sujet: Annexe 2 - Entretien avec Emmanuel Hoog

¹⁰ voir à ce sujet: Annexe 3 - Cadre légal du Dépôt Légal du Web : la DADVSI

b.Objectifs et caractéristiques

Les projets existants d'archivage du Web ont été utilisés comme inspiration afin de définir les objectifs de l'archive du Web à l'INA. Les erreurs commises et les faiblesses visibles de certains de ces systèmes ont été étudiées de manière à ce que le projet de l'INA ait une réelle valeur ajoutée par rapport aux projets existants. Une des principales différences avec des projets comme la WayBackMachine¹¹ de Internet Archive est la gestion des redondances [4]. En effet, une des principales difficultés rencontrée dans l'archivage du Web est son volume: les capacités de stockage nécessaires deviennent rapidement démesurées si la redondance des contenus archivés n'est pas gérée. Ainsi l'archive du Web de l'INA ne stocke pas en double une image lors de la collecte d'un site si l'image était déjà présente lors de la collecte précédente.

L'objectif final de l'archive du Web qui est développée à l'INA est de permettre la consultation des sites Web archivés en simulant l'interactivité du site original. Les sites sont donc collectés dans leur intégralité, avec tous les contenus « embarqués », en particulier les vidéos et les sons. Les sites sont collectés à intervalle régulier de manière à offrir aux personnes consultant l'archive plusieurs versions d'un site à des dates différentes, constituant une véritable machine à remonter dans le temps du Web.

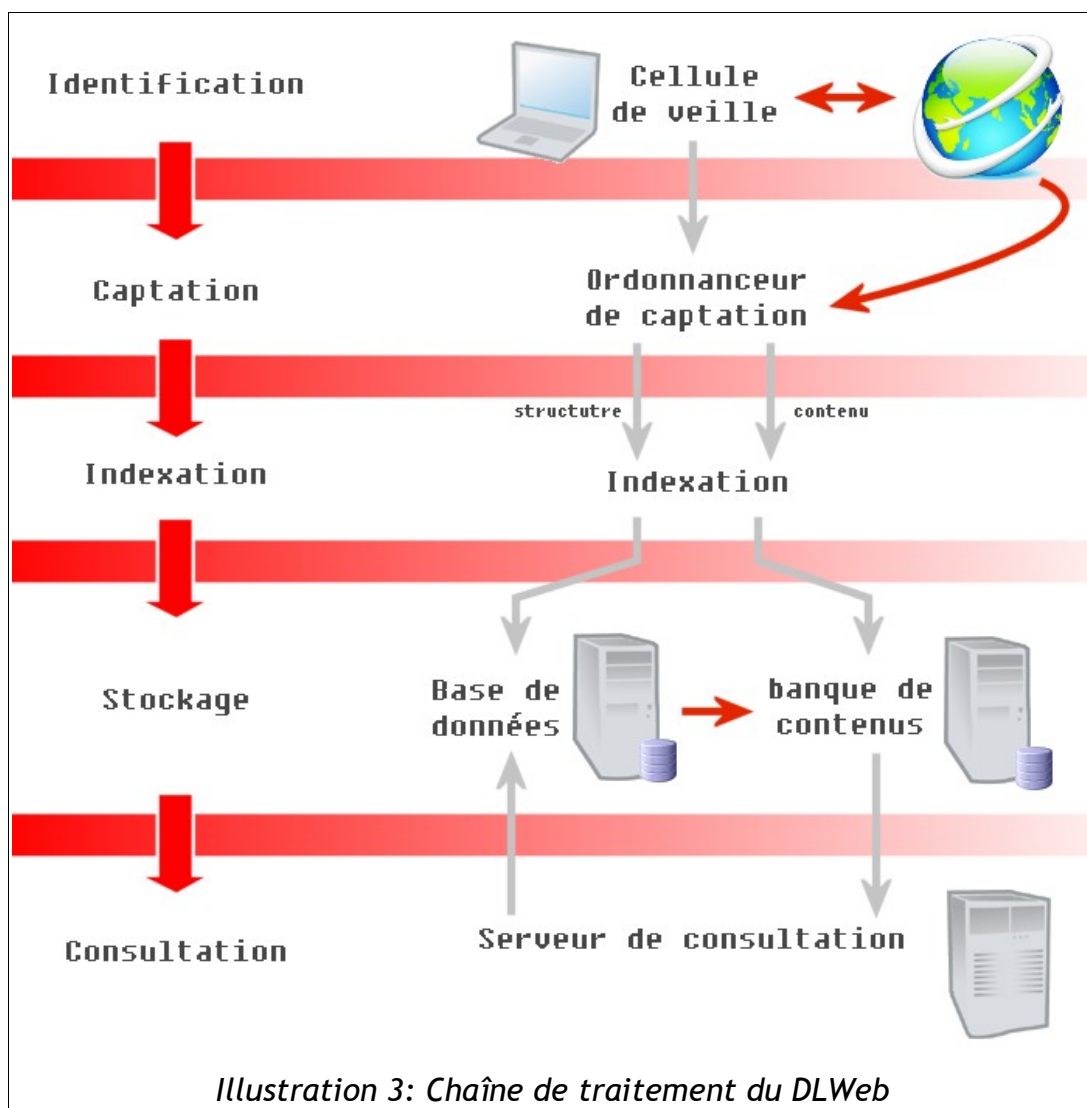
Le format de stockage utilisé doit être compatible avec un accès efficace, de manière à offrir une consultation fluide pour les usagers. Néanmoins, il doit aussi tenir compte du but initial d'une archive: la pérennité. Ainsi, le format de stockage utilisé est auto-documenté (il contient ses propres spécifications) et son contenu est agencé de manière suffisamment simple pour pouvoir être lu par un être humain sans décodage préalable.

Les sites archivés sont sélectionnés manuellement selon leur pertinence et leur intérêt pour l'archive que l'on souhaite constituer. En cohérence avec les métiers de l'INA, on s'intéressera principalement aux sites Web du milieu des industries culturelles.

11 <http://www.archive.org/web/web.php>

c. Description technique globale

Le dispositif de collecte développé à l'INA depuis 2000 vise à mettre en place un système qui réponde aux problématiques spécifiques du Web. Comme indiqué à l'illustration 3, ce dispositif est composé de 5 briques de compétence majeures.



Situé en amont de la collecte, la cellule de veille est une équipe de documentalistes chargée de prospecter le Web afin d'identifier et de suivre l'évolution des sites du domaine de recherche. La cellule de veille produit une liste de sites qui est fournie à l'ordonnanceur de captation.

L'ordonnanceur de captation est un programme chargé de planifier les campagnes de captation en établissant un calendrier adapté à la fréquence de mise à jour de chaque site. Il dirige les robots de captation qui produisent la copie d'un site Web à un instant donné.

L'indexation, semi-automatique, consiste à enrichir et documenter les informations collectées par les robots de captation pour faciliter leur recherche et leur consultation.

Les résultats indexés de la captation sont finalement stockés, en séparant les données des métadonnées, et sont rendus consultables en recréant l'interactivité des sites Web archivés via un serveur de consultation [5].

III. Projet de fin d'études

1. Objectifs

Dans le cadre du projet du Dépôt Légal du Web (DLWeb), la captation de contenus embarqués représente un enjeu de taille: en effet, l'archive a pour but d'être exhaustive du point de vue des contenus d'un site. Avec Youtube et l'avènement de la vidéo en ligne, de nombreux sites Web contiennent et diffusent maintenant des vidéos. Les procédés techniques d'intégration de ces vidéos dans les pages HTML varient mais ont la spécificité commune de ne pas rendre facilement accessible l'URL¹² de la vidéo.

Ainsi, les robots de captation ne sont pas capables d'archiver les vidéos diffusées sur une page Web selon une méthode générique. Il y a donc un besoin spécifique de donner cette compétence à l'appareil d'archivage: tel a été le sujet de mon stage.

Le sujet de stage qui m'a été initialement proposé à l'INA était le suivant: « **Conception d'outils de captation automatique de vidéo sur site web sans connaissances a priori sur la structure du site** ».

La suite de la description du stage donne le détail suivant: « **L'objectif du stage consiste en la conception, la réalisation et la documentation d'outils permettant cette capture dans deux optiques :**

- **sous forme de bibliothèques Perl réutilisables dans d'autres applications**
- **sous forme d'outils pouvant être déployés sur des postes clients standards** ».

Le développement des bibliothèques Perl concerne le Dépôt Légal du Web, présenté dans le chapitre précédant. L'outil destiné à être installé sur des postes clients standards concerne, pour sa part, le projet Signature.

Le projet Signature est un projet de recherche en charge du développement industriel d'un système permettant, grâce à une technologie développée à l'INA, de déterminer de manière automatisée et rapide si une vidéo existe dans la base de connaissances dudit système. Cette technologie est notamment utilisée pour déterminer si une vidéo diffusée appartient à l'ensemble des vidéos archivées à l'INA. Si c'est le cas, l'INA se charge d'identifier et de rétribuer les ayants-droits.

Dans le cadre de la détection de vidéos archivées à l'INA et diffusées illégalement (sans paiement de droits de diffusion), le projet Signature s'intéresse à la possibilité de réaliser une telle détection sur les vidéos diffusées via le Web. Ainsi le service juridique de l'INA, en charge de la résolution des conflits de droit de diffusion, a souhaité le développement d'un outil permettant à une personne non-experte de télécharger sur son ordinateur une vidéo diffusée sur une page Web. L'objectif du développement d'un tel outil est de pouvoir, à partir de l'adresse d'une page Web contenant une vidéo, télécharger cette vidéo et la soumettre au test du système de détection.

¹² Uniform Resource Locator: système d'adressage de contenus sur Internet. Par extension, l'URL d'un contenu est son adresse sur Internet.

2. Rôles occupés dans les projets

Pour la partie de mon stage concernant le projet du Dépôt Légal du Web, mon rôle a été de me joindre à Thomas Drugeon et Nicolas Delaforge, qui sont chargés respectivement du développement du dispositif de captation, de stockage et de consultation; et de l'interface de consultation.

Mon travail s'est essentiellement déroulé aux côtés de Thomas Drugeon qui est l'ingénieur responsable du projet. Son expertise des librairies Perl du projet est très élevée puisqu'il en est l'auteur principal. C'est donc guidé par ses conseils et en utilisant le framework de programmation asynchrone dont il est l'auteur que j'ai développé une librairie Perl permettant de retrouver et de télécharger les vidéos affichées dans une page Web. La place de cette librairie dans le workflow d'archivage sera discutée par la suite.

La partie de mon stage concernant le projet Signature s'est déroulée sous la direction et avec la collaboration de Frédéric Dumas. Après avoir précisé et analysé avec lui le besoin exprimé dans le sujet de stage, j'ai travaillé sur ce projet en parallèle du projet du Dépôt Légal du Web. Mon bureau étant situé au DLWeb, j'ai participé à une grande partie des réunions quotidiennes de l'équipe Signature afin de la tenir informée des mes avancées et de mes difficultés durant le développement de l'outil.

Mon rôle a donc été d'effectuer un travail d'intégration de nouvelles librairies dans un système déjà complexe et avancé d'une part, et de développer un outil neuf indépendant du projet global, répondant à un besoin localisé d'autre part.

3. Antécédents nécessaires

Voici en bref les prérequis aux travaux que j'ai réalisés dans le cadre de ce stage.

Pour commencer, la connaissance du langage de programmation Perl que j'ai eu l'occasion d'apprendre sommairement lors de travaux expérimentaux pour le département Génie des Systèmes Mécaniques à l'UTC au printemps 2006 m'a été très utile.

Ensuite, la connaissance des technologies du Web en général (HTML, HTTP, Flash), m'a permis d'avancer rapidement dans mon travail d'analyse.

Enfin, de nombreuses choses ont dû être approfondies ou tout simplement abordées au cours de mon stage. En effet, il m'a fallu étudier comment les différents hébergeurs de vidéos sur le Web organisent et protègent l'accès à leurs vidéos. J'ai également dû apprendre comment développer une extension Firefox pour réaliser l'outil déployable sur des postes clients standards à l'INA.

4.Plannings du travail

Voici le planning effectif du semestre de stage. Mon travail sur le projet DLWeb et le projet Signature, parfois simultanément, à été indiqué dans des couleurs différentes afin de mettre en relief l'importance relative de chaque projet.

Mois	Partie 1	Partie 2	Partie 3	Partie 4
Février	Découverte de l'INA, formalités administratives.	Découverte du framework de programmation utilisé.	Expérimentation et découverte des outils existants.	
Mars	Développement d'un prototype spécialisé dans Youtube et Dailymotion.	Formalisation des méthodes d'accès aux vidéos.	Amélioration des stratégies de capture.	Réflexion sur l'architecture globale de l'application.
Avril	Réécriture de l'application avec l'architecture choisie, permettant la modularité de l'application.		Tests de l'application réécrite et réparation des bugs détectés. Nombreuses collectes autour de la campagne présidentielle.	
Mai	Collecte intensive autour de la campagne présidentielle. Ajout de nouvelles stratégies-site.	Étude de l'existant et apprentissage de la programmation Firefox.	Développement et debug de l'extension Firefox.	Tests et collectes de vidéos avec la librairie Perl.
				Modification de l'extension Firefox suite au retour utilisateur.
Juin	Ajout d'une interface de monitoring pour la librairie Perl.	Modification de l'architecture pour améliorer les possibilités de monitoring.	Debug de la librairie Perl. Modification de l'expression régulière d'extraction d'URL de vidéos.	Collecte de vidéos et modification de certaines stratégies.
				Ajout du support du streaming à l'extension
Juillet	Rédaction du rapport de stage et collectes de vidéos à l'aide de la librairie Perl. Collectes avec la librairie. Correction de bugs et développement. Rédaction de la documentation.			/

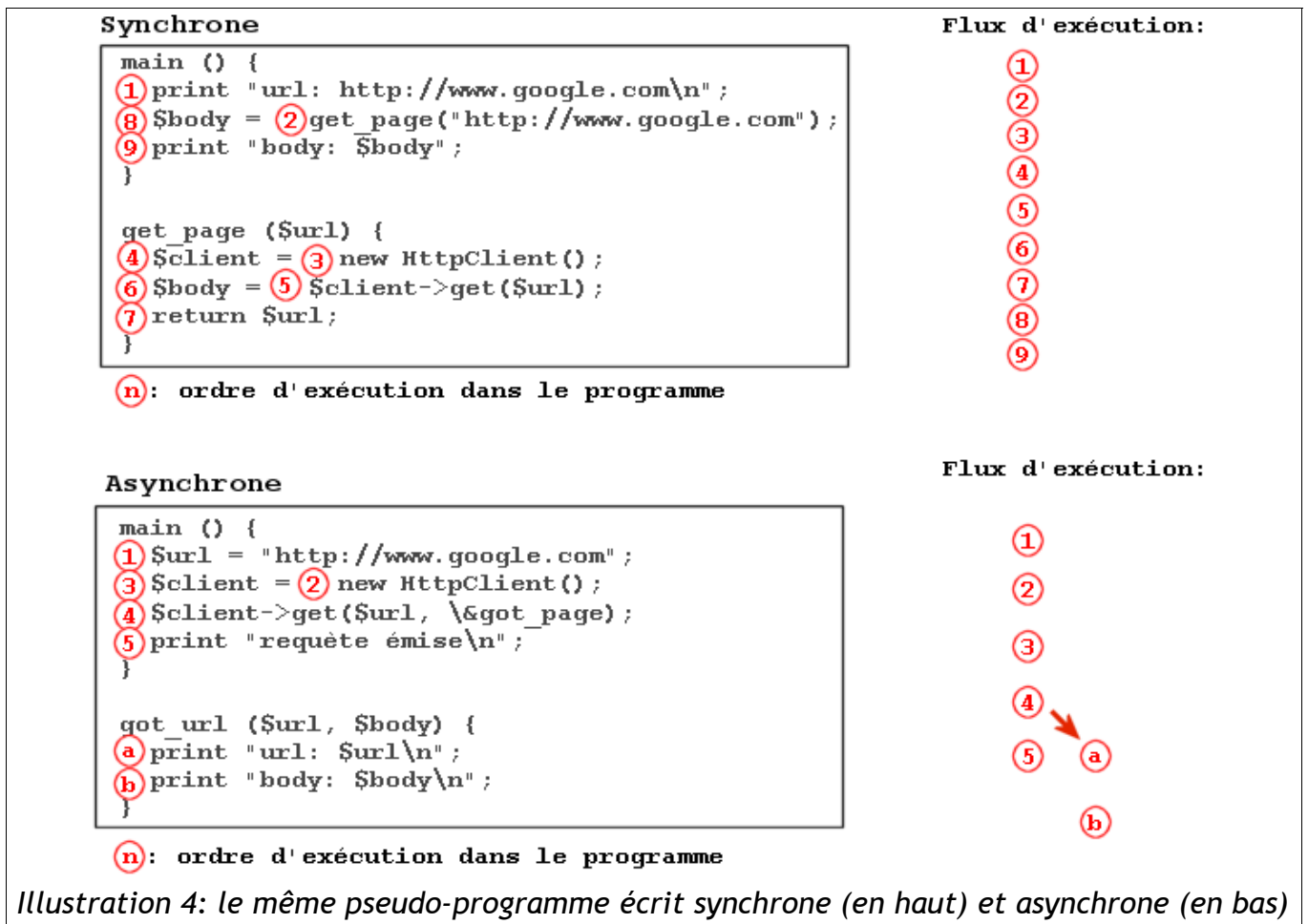
Travail dans le cadre du projet Signature / Travail dans le cadre du projet DLWeb

5. Librairie de capation en Perl : VideoScrap

a. Compréhension du framework et des outils de développement

La grande majorité des bibliothèques développées au DLWeb sont écrites pour être utilisées dans un framework de programmation asynchrone : **Anet** (Asynchronous Networking). Développé à l'INA dans le cadre de ce projet, ce framework asynchrone à *Thread* unique a pour but de permettre la gestion de processus fortement multitâches en réseau sans l'encombrement mémoire qu'induit l'utilisation de *Threads*. En effet la souplesse d'utilisation d'un langage interprété¹³ et à gestion automatique de la mémoire tel que Perl se fait au prix d'efforts importants pour économiser la mémoire vive et le temps machine.

Anet est un système asynchrone, cela signifie que lorsqu'une fonction est appelée par une portion de code, l'exécution de la portion de code appelante n'est pas interrompue au profit de celle de la fonction appelée. Si on désire recevoir une réponse ou des données de la part de la fonction appelée, on doit lui fournir un nom de fonction à appeler pour passer des données.



Il y a donc, comme indiqué à l'illustration 4, des exécutions « simultanées ». Comme l'architecture d'Anet est à *Thread* unique, une boucle centrale à Anet passe la main d'une fonction à une autre à la manière d'un ordonnanceur. Les systèmes asynchrones peuvent dans un premier temps paraître plus difficiles à comprendre et à corriger car le flux de l'exécution

13 Par opposition à un langage compilé, comme le C++

ne semble pas linéaire, rendant plus complexe la localisation des erreurs dans le code source.

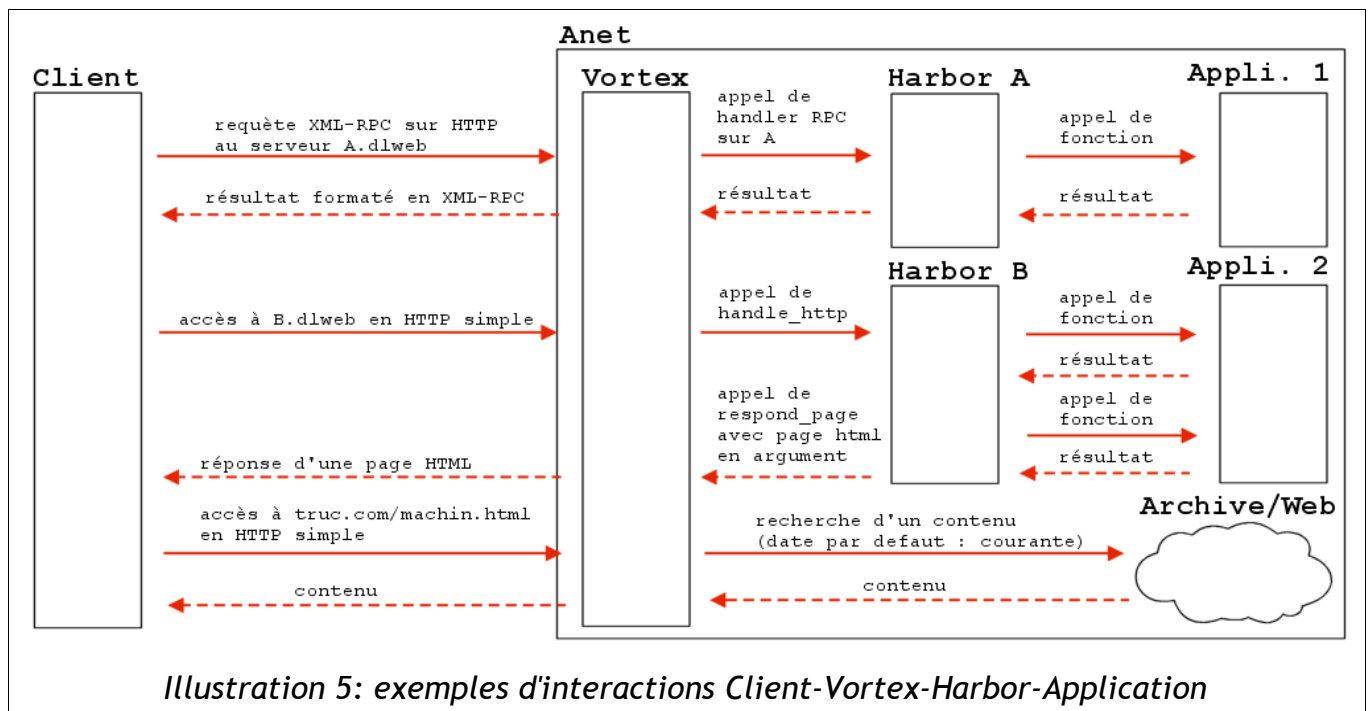
Autour d'Anet a été développé toute une série de bibliothèques asynchrones, dont notamment un client HTTP qui est utilisé au cœur de la bibliothèque que j'ai développée. Les programmes écrits dans le framework Anet peuvent être exécutés en ligne de commande avec un interpréteur Perl ou bien par l'intermédiaire d'un proxy d'applications développé au DLWeb : le Vortex (initialement développé en tant que serveur d'archive).

Un **Vortex** est un programme écrit dans le framework Anet qui, exécuté sur une machine, joue le rôle de proxy HTTP dans le réseau auquel est connectée cette machine. Il « écoute » donc les messages reçus sur un port donné et émis dans le protocole HTTP. Les messages reçus peuvent être des messages HTTP simples ou bien formatés en SOAP¹⁴, XML-RPC, JSRS¹⁵ ou PPC¹⁶. Son rôle initial est de rediriger les requêtes HTTP qu'il reçoit vers l'archive. Lorsqu'un Vortex est initialisé, un certain nombre d'applications peut s'enregistrer auprès de lui. Lorsqu'un message est reçu, le Vortex le décode et effectue la redirection vers l'archive ou bien l'appel de méthode correspondant auprès de l'application concernée le cas échéant [5].

De manière à faciliter l'utilisation d'une application via un Vortex, une interface entre les applications et un Vortex existe : le **Harbor**. Un Harbor est une classe qui rassemble et présente un certain nombre de fonctionnalités à travers un Vortex. Les fonctionnalités présentées par un Harbor appartiennent habituellement à une seule application. Lorsque le Vortex reçoit une requête HTTP, c'est le serveur indiqué dans l'URL qui lui permet de déterminer le Harbor destinataire de la requête.

Par exemple, si un client veut effectuer un accès HTTP simple auprès du Harbor B, il lui suffit d'indiquer l'IP du Vortex comme proxy de son navigateur Web et d'accéder au site *B.dlweb*. Le Vortex va recevoir la requête, reconnaître l'adresse se terminant en « .dlweb » et effectuer un appel à la méthode *handle_http* du Harbor B (voir illustration 5). La réponse (vraisemblablement une page web) sera renvoyée par le Vortex et affichée par le navigateur comme si *B.dlweb* était effectivement un site Web.

Les adresses autres que « *.dlweb » sont restitués depuis l'archive du Web directement.



14 Protocole de requêtes distantes (RPC) formatées en XML

15 JavaScript Remote Scripting : une technique pour créer de applications Web interactives, similaire à AJAX

16 Path Procedure Call: un protocole permettant de faire des appel de méthodes formatés en URLs (sur HTTP)

b. Présentation des procédés effectifs d'intégration de vidéos

L'intégration de vidéos dans des pages Web telle que je devais la traiter dans le cadre de ce projet concernait les intégrations utilisant Flash. En effet, depuis l'avènement de Youtube et de ses nombreux clones, la méthode la plus populaire pour intégrer une vidéo dans une page Web est de le faire via un lecteur Flash. La préférence pour cette solution a de multiples causes.

Tout d'abord, l'implantation de Flash dans le monde du Web est très bonne: sa popularité est liée à la facilité de développer dans ce langage. Le plugin Flash, permettant d'interpréter des programmes Flash, est compatible avec tous les navigateurs populaires (Internet Explorer, Mozilla Firefox, Opera, Safari) et sur la plupart des plateformes (Windows, Mac, Linux). Cette maturité d'intégration et la demande croissante de vidéos en ligne, matérialisée par l'apparition spontanée de nombreux sites proposant de courtes vidéos, souvent diffusées en format WMV ou ASF (Windows Média en format statique ou streamé) a poussé les développeurs de Flash à rendre leur logiciel capable de jouer de la vidéo en pseudo-streaming. Cette fonctionnalité existait déjà pour diffuser des fichiers MP3 sans que leur téléchargement intégral soit nécessaire pour commencer la lecture. Les limitations techniques des codecs vidéos actuels de ce point de vue ont été à l'origine du développement d'un codec vidéo adapté à la lecture de fichiers partiellement téléchargés.

Avec le pseudo-streaming, pour les diffuseurs, il n'est plus nécessaire d'installer un serveur dédié à la diffusion de vidéos car les fichiers vidéo ont simplement à être rendus disponibles dans le format FLV sur un serveur de fichiers HTTP. Aussi, contrairement au streaming « pur », le pseudo-streaming télécharge au débit maximal et dans son intégralité le fichier à lire sur la machine du lecteur, ce qui évite les ralentissements de lecture connus avec le streaming « pur », où la vidéo est partiellement téléchargée dans un tampon dont la faible taille nécessite des protocoles complexes et la sollicitation importante des serveurs afin d'assurer la diffusion en continu.

La simultanéité de cette nouvelle possibilité technique avec la généralisation de débits de connections personnelles permettant son utilisation à grande échelle n'est certainement pas un hasard, et son apparition a très rapidement séduit un certain nombre de développeurs de sites Web qui ont commencé à l'utiliser massivement à partir de 2005 ... année de la création du site Youtube.

L'intégration d'un objet Flash dans une page HTML se fait généralement par l'utilisation d'une combinaison de balises « <OBJECT> », « <PARAM> » et « <EMBED> » (voir illustration 6).

```
<object width="425" height="350">
  <param
    name="movie" value="http://www.youtube.com/v/xnujO3SCGBE">
  </param>
  <param
    name="wmode" value="transparent">
  </param>
  <embed
    src="http://www.youtube.com/v/xnujO3SCGBE"
    type="application/x-shockwave-flash"
    wmode="transparent">
  </embed>
</object>
```

Illustration 6: exemple d'intégration d'un objet Flash en HTML

Cette combinaison de balises sert à indiquer au navigateur Web l'URL de l'objet à intégrer dans la page, grâce à l'attribut « SRC », ainsi que le type MIME¹⁷ du contenu, grâce à l'attribut « TYPE » de la balise « <EMBED> ». De plus, les balises « <PARAM> » servent à indiquer une série de couples paramètre/valeur qui pourront être utilisés à l'intérieur du programme Flash. Ce programme Flash se présente sous la forme d'un fichier SWF qui peut être exécuté par un interpréteur (ou plugin) Flash au sein d'un navigateur.

Il est important de noter que le programme Flash doit recevoir, de la part de la page qui le contient, les informations nécessaires pour accéder au fichier de la vidéo qu'il devra lire. L'adresse de la vidéo peut être donnée directement, ou bien des informations permettant de reconstruire l'adresse de la vidéo sont transmises. Il faut noter qu'un programme Flash est capable de communiquer en HTTP, il peut donc faire des requêtes et des traitements complexes afin de retrouver l'URL réelle de la vidéo à lire.

Aussi, il n'est pas toujours suffisant de rechercher les balises « <EMBED> » dans le code HTML d'une page pour en découvrir toutes les vidéos. En effet, la portion de code HTML présentée à l'illustration 6 peut être ajoutée à la structure de la page dynamiquement en JavaScript, après le chargement du document HTML. Cette technique de génération dynamique de page HTML, devenue classique avec le Web 2.0, est notamment utilisée afin de rendre plus compliquée la tâche des robots qui tentent d'effectuer l'analyse syntaxique du code d'une page. Il est donc parfois nécessaire d'analyser le code JavaScript d'une page afin d'extraire les informations des portions qui modifient dynamiquement le contenu de la page.

Le contexte technique de développement fait que les combinaisons de méthodes décrites précédemment ainsi que l'apparition de nouvelles méthode interdisent, mêmes aux imaginations les plus fertiles, de concevoir une technique tout à fait générique permettant de détecter l'existence d'une vidéo dans une page Web et de la télécharger. Un échantillon représentatif des méthodes mises en oeuvre par les hébergeurs de vidéos pour accéder à leur contenus va maintenant être détaillé afin de comprendre les motivations des choix architecturaux qui vont être décrits par la suite.

Commençons par le site de vidéos en ligne le plus populaire de la planète à ce jour : Youtube. Il existe principalement deux moyens de voir une vidéo Youtube: sur le site Youtube lui-même, ou bien sur une page d'un autre site (voir illustration 7).



Illustration 7: vidéos Youtube sur Youtube.com et sur LeMonde.fr

17 Multipurpose Internet Mail Extensions (MIME) est un standard internet qui décrit le format de données

Sur le site Youtube.com, l'intégration de la vidéo est faite dynamiquement en JavaScript. Dans les autres sites, l'intégration se fait avec la portion de code HTML présentée à l'illustration 6 p.19. Dans le cas de l'intégration sur d'autres sites que Youtube, l'adresse du contenu embarqué donnée est de la forme <http://www.youtube.com/v/6VdNcCweL0>. Au chargement de la page, le navigateur va donc faire une requête pour lire le contenu disponible a cette adresse. Cette adresse va en réalité s'avérer être une redirection HTTP, ce qui signifie qu'au lieu d'un contenu, elle correspond à une autre adresse à visiter :

http://www.youtube.com/jp.swf?video_id=6VdNcCweL0&eurl=http%3A//www.lemonde.fr/web/article/0%2C1-0%402-3222%2C36-930388%2C0.html&iurl=http%3A//img.youtube.com/vi/6VdNcCweL0/default.jpg&t=OEgsToPDskL3UP9SHPb0arK324ASUwhj

La première adresse n'est donc qu'une simplification qui renvoie vers le contenu attendu: un programme Flash (SWF). Dans l'URL finale, nous pouvons constater que plusieurs paramètres informent le programme Flash à propos de la vidéo à lire. L'identifiant de la vidéo lui est passé (**video_id**) ainsi que l'adresse de l'imagette à afficher avant le début de la lecture (**iurl**), l'adresse de la page contenant cette vidéo (**eurl**) et enfin, une clé de session (**t**). Des outils de monitoring appropriés¹⁸ permettent de constater que le programme Flash accède à l'adresse suivante au moment du chargement :

http://www.youtube.com/get_video?video_id=6VdNcCweL0&t=OEgsToPDskJWZ91QZOT3uJtjbm f8OSu2&eurl=http%3A%2F%2Fwww%2Emonde%2Efr%2Fweb%2Farticle%2F0%2C1%2D0%402%2D3222%2C36%2D930388%2C0%2Ehtml

A décomposer cette requête, on découvre qu'elle a été construite en rassemblant les paramètres **video_id**, **t**, **eurl** et la racine http://www.youtube.com/get_video. Lorsqu'on accède à cette requête manuellement, on constate qu'on télécharge un fichier qui n'est autre que la vidéo au format FLV. Notre objectif sera donc d'enregistrer ce cheminement et les données qu'il contient, de manière à pouvoir le « rejouer » dans l'archive.

Voyons à présent le cas de MySpace, le très populaire hébergeur de blogs, qui s'est mis au goût de la vidéo sur internet récemment. Les vidéos hébergées par MySpace peuvent être visionnées sur la plateforme vidéo de MySpace à l'adresse vids.myspace.com mais aussi sur d'autres sites, y compris sur des blogs MySpace. Lorsque les vidéos sont intégrées à d'autres sites que vids.myspace.com, elle le sont avec une portion de code dont la partie caractéristique est la suivante:

```
<embed src="http://lads.myspace.com/videos/vplayer.swf"
flashvars="m=2011389529&type=video" />
```

On constate ici que beaucoup moins d'informations sont transmises au programme Flash: seulement deux paramètres dont un (**type**) indique le type de média, et un second (**m**) un numéro qui sert à identifier la vidéo. Lorsqu'on accède à cette vidéo via un navigateur Web, on constate que le programme Flash effectue une requête à l'URL suivante:

<http://mediaservices.myspace.com/services/rss.ashx?type=video&mediaID=201138952>

Cette URL se construit avec les deux paramètres passés au programme Flash (où **m** est devenu **mediaID**) ajoutés à une racine fixe. Si on visite cette URL avec un navigateur, on obtient un fichier de métadonnées formatées en XML concernant la vidéo (voir illustration 8 p.22). Parmi les métadonnées reçues, on peut isoler l'URL du fichier vidéo qui nous intéresse (champ **media:content**) ainsi que celle de l'imagette à afficher avant le début de la lecture (champ **media:thumbnail**). Ces deux contenus ainsi que toutes les étapes intermédiaires qui y ont mené seront archivés.

18 L'extension Tamper Data pour Firefox permet d'observer toutes les requêtes émises par le navigateur et ses plugins, y compris Flash.

```

<?xml version="1.0" encoding="utf-8"?>
<rss version="2.0" xmlns:media="http://search.yahoo.com/mrss"
xmlns:myspace="http://myspace.com/">
  <channel>
    <title>MySpace</title>
    <description>MySpace</description>
    <link>http://myspace.com/50872026</link>
    <lastBuildDate>Wed, 04 Jul 2007 05:06:52 GMT</lastBuildDate>
    <docs>http://myspace.com</docs>
    <generator>ELS2MWEBNET0619</generator>
    <myspace:friendID>50872026</myspace:friendID>
    <item>
      <title>88 (TOKYO REMIX)</title>
      <pubDate>Wed, 04 Jul 2007 05:06:52 GMT</pubDate>
      <media:thumbnail url="http://myspace-
529.vo.llnwd.net/02011/92/59/2011389529_thumb0.jpg" />
      <media:content
url="http://content.movies.myspace.com/0020113/92/59/2011389529.flv"
type="video/x-flv" medium="video" duration="252" />
      <myspace:mediaID>2011389529</myspace:mediaID>
      <myspace:mediaSource>Video</myspace:mediaSource>
      <myspace:itemID>2011389529</myspace:itemID>
      <myspace:itemType>Video</myspace:itemType>
      <myspace:friendID>50872026</myspace:friendID>
    </item>
  </channel>
</rss>

```

Illustration 8: métadonnées d'une vidéo MySpace

Les exemples donnés pour Youtube et MySpace ne représentent que deux manières particulières d'accéder à des vidéos hébergées. Chez ces mêmes hébergeurs ainsi que chez les autres existants et à venir, de nombreux autres cas existent. Pour chacun de ces cas, il faut isoler les informations nécessaires pour être capable de simuler la séquence de requêtes émises par le navigateur et le programme Flash. L'examen des deux cas précédents démontre que dans la plupart des étapes, il est impossible d'imaginer une stratégie générique pour extraire les informations pertinentes et passer à l'étape suivante. L'architecture de la librairie devra donc contenir un ensemble de stratégies spécialisées pour chacun des cas connus, et si possible robuste aux variations sur un même cas.

c. Architecture et implémentation de la librairie

Nous allons aborder ici une partie technique, qui tente de rendre compte du fonctionnement de l'application, du rôle et des interactions de chacune de ses briques. J'essaierai autant que possible de justifier les choix d'architecture et d'illustrer la pertinence de ces choix. Les explications qui vont suivre seront plus facilement compréhensibles en ayant à l'esprit le diagramme de classes de l'application en annexe 4.

Comme nous l'avons vu dans la partie précédente, il existe pour chaque cas d'intégration de vidéo une séquence d'URLs à visiter, chaque URL contenant les informations nécessaires pour atteindre la suivante. De manière à ce que la librairie s'intègre au mieux dans le système actuel de collecte, nous avons décidé que n'importe quelle URL d'une séquence pouvait être un point d'entrée. De cette manière, si pour obtenir la vidéo présente à l'adresse D, il faut passer par les adresses A, B et C alors il faut que A, B et C soient des points d'entrée possibles.

Pour ce faire, les URLs données à traiter passent par une succession d'« aiguillages » afin de les reconnaître et de leur appliquer le traitement adéquat. La classe *Strategy*, qui correspond en fait à l'ensemble des stratégies connues pour un site donné, possède un *dispatcher* lui permettant de choisir la fonction de traitement adaptée à une URL. Afin de pouvoir donner une URL à la *Strategy* qui saura la traiter, une *Strategy* particulière est utilisée : *Smart*. *Smart* extrait le nom de domaine¹⁹ de l'URL et détermine la *Strategy* à utiliser grâce à une liste de correspondances nom de domaine/*Strategy*. Une fois la *Strategy* appropriée choisie, *Smart* ré-aiguille l'URL.

Afin de faciliter le ré-aiguillage des URLs à travers le programme et de stocker des informations de contexte avec elle, un objet *Transaction* est utilisé. Cet objet représente une vidéo à télécharger et va stocker toutes les URLs de la séquence menant à cette vidéo, ainsi que le contenu des pages de cette séquence. Une trace des choix d'aiguillage va également être stockée afin de faciliter le debuggage. Il arrive parfois qu'une URL corresponde à plusieurs vidéos, par exemple si l'URL référence une page de blog contenant plusieurs vidéos embarquées. Pour cette raison et dans un soucis d'homogénéité avec le nombre d'URLs entrées, l'objet *Entry* est utilisé. Une *Entry* représente une URL qui a été entrée dans le programme, et contient une ou plusieurs *Transactions*.

Voyons le chemin parcouru par une *Transaction* d'une URL du site Dailymotion:

1. entrée de <http://www.dailymotion.com/swf/h3xNiXGHfNqIegUJe>
2. création d'une *Entry* contenant une *Transaction*
3. routage de la *Transaction* dans la stratégie *Smart* (par défaut)
4. détection d'une URL Dailymotion par *Smart*
5. routage de la *Transaction* vers la stratégie Dailymotion
6. détection d'une URL « redirection vers vidéo embarquée »
7. lecture du contenu à cette URL pour obtenir la redirection attendue
8. traitement de la redirection vers
<http://dailymotion.com/flash/flvplayer.swf?rev=...>
9. extraction de l'adresse finale de la vidéo, présent dans cette URL
10. téléchargement du fichier vidéo
11. stockage du fichier vidéo et de ses métadonnées
12. destruction de la *Transaction*, puis de l'*Entry*, mise à jour des statistiques

Illustration 9: traitement d'une URL Dailvmotion

¹⁹ Le nom de domaine d'une URL est le nom du serveur dont on a gardé que les deux dernières composantes. Par exemple le nom de domaine de l'URL <http://truc.machin.fr/roflmao.html> est *machin.fr*

Ce chemin illustre le flux des transactions au sein d'une *Strategy* ainsi que d'une *Strategy* à l'autre, en particulier au moment du routage par *Smart*. A l'illustration 10, il est important de noter que si *Smart* ne trouve aucune *Strategy* capable de traiter la *Transaction*, il décide de la router vers une autre *Strategy* particulière : *Embedded*. Cette *Strategy* part du constat que l'URL n'a pas permis de déterminer la marche à suivre. Il s'agit donc très vraisemblablement d'une page d'un site n'étant pas un site connu mais contenant éventuellement des vidéos venant de sites connus. Afin de permettre à chaque *Strategy* de rechercher des vidéos à traiter à cette adresse, le corps de la page est enregistré et transmis à toutes les *Strategy* connues, afin de leur permettre d'analyser le corps de la page à la recherche de vidéos embarquées. Contrairement à *Smart*, qui route une *Transaction* vers une seule destination, *Embedded* duplique la *Transaction* autant de fois qu'il existe de *Strategies* et l'envoie à toutes. On voit ici l'utilité d'avoir un objet *Entry* qui regroupe toutes les *Transactions* étant issues de la même URL d'origine.

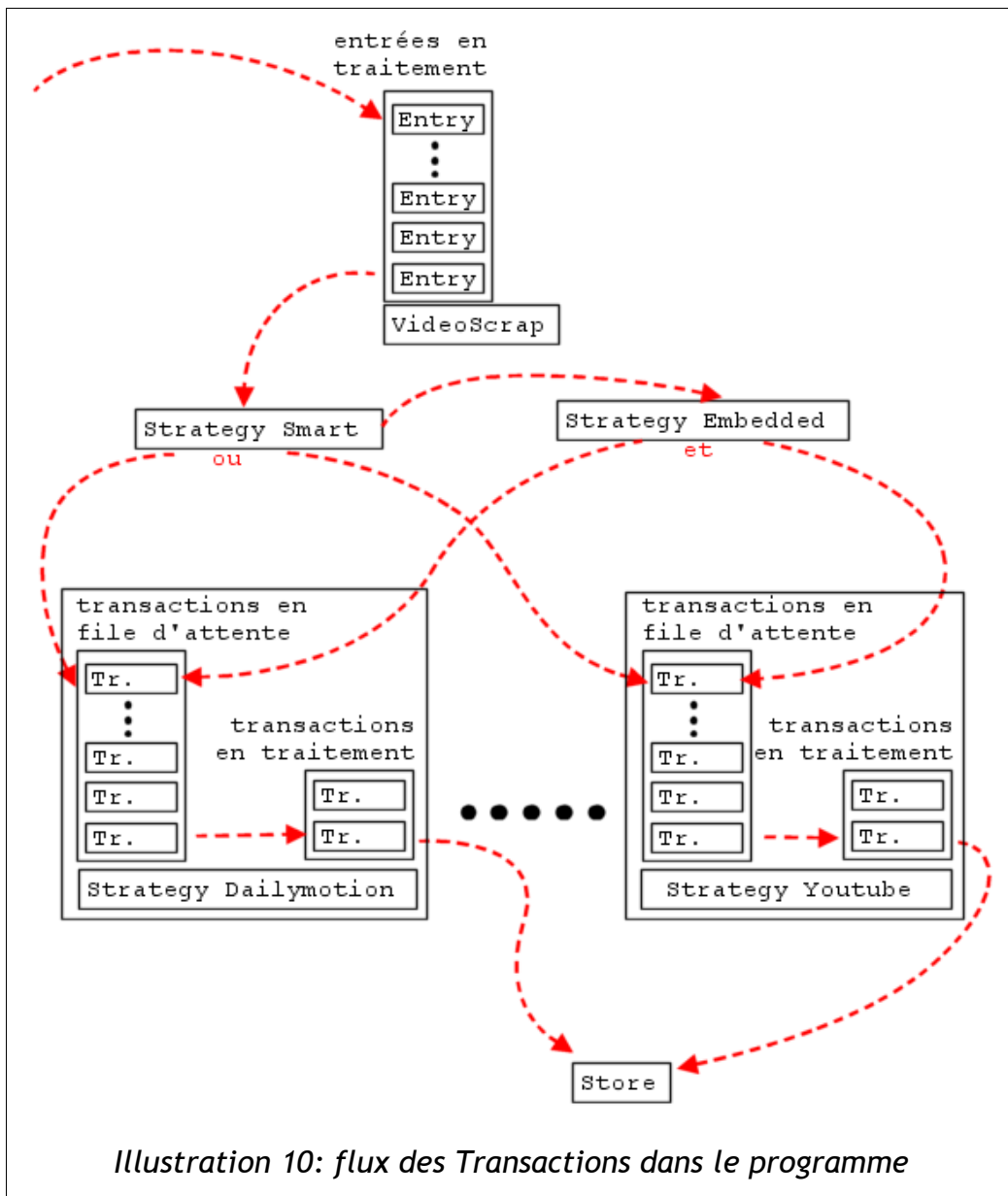


Illustration 10: flux des Transactions dans le programme

Après le routage, nous abordons la question du stockage. En effet, une des problématiques de l'archivage du Web qui a été abordée est le volume de données. Le volume important du Web est en grande partie dû aux doublons et aux références multiples: un contenu peut être présent à l'identique sur plusieurs serveurs ou bien un contenu unique peut

être référencé par plusieurs pages. Dans un premier temps nous avons tenté d'améliorer le traitement du cas d'une vidéo déjà téléchargée. La problématique est la suivante: un contenu identique peut être pointé par des URLs différentes, nous voulons archiver toutes les URLs menant à un même contenu mais souhaitons éviter de télécharger le même fichier plusieurs fois. Pour cela, nous avons décidé de tirer parti de l'identification de fichier qui est utilisée en interne par chacun des sites que nous traitons. Ainsi, et comme cela est visible dans les exemples précédents (voir illustration 8 p.22: `mediaID`), les hébergeurs de fichiers vidéos utilisent en interne des identificateurs de fichiers afin d'interroger leurs bases de données; l'identificateur de fichier vidéo peut apparaître plus ou moins tôt dans la séquence de pages menant à un fichier vidéo, mais est toujours présent. La tactique mise en oeuvre consiste à stocker avec chaque vidéo l'information concernant son site d'origine et son identificateur sur ce site. Ainsi, dès que l'identificateur de vidéo est détecté dans une séquence, les fichiers déjà téléchargés sont interrogés afin de savoir si la vidéo est déjà présente parmi eux. Si c'est le cas, la phase de téléchargement du fichier sera sautée et seul les méta-données de la vidéo seront enregistrées.

Le cas de fichiers venant de sites différents ou ayant des identificateurs différents mais étant tout de même identiques par leur contenu est traité en aval du téléchargement par la librairie. Lors de la consolidation de l'archive²⁰, les contenus ajoutés sont signés²¹ et comparés aux contenus déjà présents dans l'archive consolidée. Lors de la mise en évidence d'un doublon, celui-ci n'est pas copié. Cette technique permet de ne pas stocker en double dans l'archive finale les données dont l'identité n'était pas détectable à priori.

Les mécanismes de stockage présentés ci-avant sont génériques. Dans la pratique, il faut indiquer au programme à son initialisation quelle classe de stockage utiliser. Une classe de stockage (*Store*) doit implémenter une fonction permettant de stocker le contenu sur disque et une fonction permettant de vérifier l'existence du contenu dans les fichiers déjà téléchargés. Il existe différentes implémentations de la classe de stockage, permettant notamment d'écrire des fichiers simples, d'écrire directement dans des fichiers d'archive DAFF, ou d'écrire les données sur un serveur distant via *ssh*.

La présentation de la logique de fonctionnement qui vient d'être faite n'entre pas dans le détail de certaines utilisations spécifiques et ne couvre pas tout le champ des possibilités offertes par l'application. Il s'agit plus ici de donner les clés du fonctionnement que de passer en revue les détails d'implémentation et de configuration.

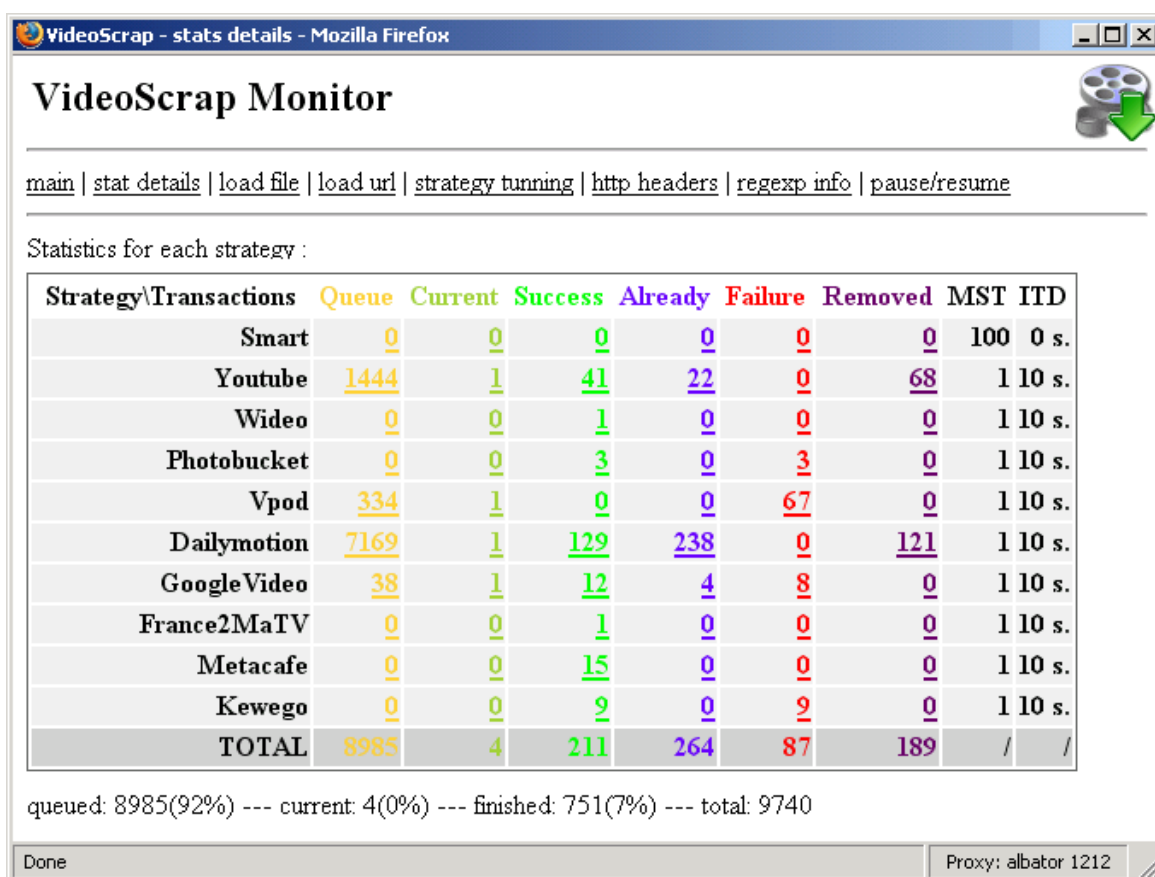
20 Lors de cette phase, également appelée phase de *versement*, les données récemment collectées sont ajoutées au contenu indexé et consultable de l'archive.

21 La signature d'un contenu est une opération injective qui permet de produire à partir d'un contenu de longueur quelconque un code de taille fixe « caractéristique » du contenu codé. Le but d'un tel mécanisme est de donner un « résumé » du contenu ayant une très faible possibilité de collision avec le « résumé » d'un contenu différent. La signature utilisée ici est de type SHA256.

d. Interface de monitoring et de contrôle

Afin de pouvoir utiliser facilement la librairie qui a été décrite dans les parties précédentes et de monitorer simplement son fonctionnement et ses éventuels dysfonctionnements, une interface a été développée. Cette interface se présente sous la forme d'un *Harbor*²² et est donc consultable via un navigateur Web. L'utilisation actuelle de la librairie est de traiter des fichiers contenant une liste d'URLs de vidéos. L'interface permet de sélectionner un fichier de ce type et de le charger. Une fois chargé, le fichier est lu ligne par ligne et chaque URL est extraite pour être traitée. La possibilité est également donnée de fournir les URLs à traiter via des appels XML-RPC au Harbor, permettant ainsi l'interaction avec un autre programme²³.

L'interface Web du Harbor permet d'afficher un tableau de bord de l'évolution de la captation afin de donner à l'utilisateur une idée de la progression de la captation, du nombre d'erreurs survenues, du nombre de vidéos déjà connues rencontrées, du nombre de vidéos nouvelles téléchargées, ainsi que du nombre de vidéos qui ont été définitivement mises hors lignes et qui n'ont pu être téléchargées (voir illustration 11).



The screenshot shows a web browser window titled "VideoScrap - stats details - Mozilla Firefox". The page title is "VideoScrap Monitor". There are navigation links: [main](#), [stat details](#), [load file](#), [load url](#), [strategy tuning](#), [http headers](#), [regexp info](#), and [pause/resume](#). Below the links, it says "Statistics for each strategy :".

Strategy\Transactions	Queue	Current	Success	Already	Failure	Removed	MST	ITD
Smart	0	0	0	0	0	0	100	0 s.
Youtube	1444	1	41	22	0	68	1	10 s.
Wideo	0	0	1	0	0	0	1	10 s.
Photobucket	0	0	3	0	3	0	1	10 s.
Vpod	334	1	0	0	67	0	1	10 s.
Dailymotion	7169	1	129	238	0	121	1	10 s.
GoogleVideo	38	1	12	4	8	0	1	10 s.
France2MaTV	0	0	1	0	0	0	1	10 s.
Metacafe	0	0	15	0	0	0	1	10 s.
Kewego	0	0	9	0	9	0	1	10 s.
TOTAL	8985	4	211	264	87	189	/	/

queued: 8985(92%) --- current: 4(0%) --- finished: 751(7%) --- total: 9740

Done Proxy: albator 1212

Illustration 11: tableau de bord principal de l'interface de monitoring

Chaque ligne correspond à une *Strategy* et les chiffres indiqués sont un nombre de *Transactions*. Des informations complémentaires, comme le nombre maximum de transactions simultanées (MST) ou le délai inter-transaction (ITD), sont indiquées afin de permettre à l'utilisateur de comprendre la raison d'une éventuelle lenteur apparente. Ces valeurs dites de « tuning » des transactions (MST et ITD) sont modifiables par le biais du

22 cf. page 18

23 On pense ici notamment aux robots de collecte de sites Web

menu du même nom. Ainsi l'utilisateur peut choisir de moduler la fréquence et l'intensité de la collecte en fonction des précautions qu'il souhaite prendre. Ce type de réglages peut permettre d'éviter au robot de surcharger un site et d'en être banni en réponse. Comme indiqué à l'illustration 12, chaque réglage se fait au niveau d'une stratégie, ce qui est équivalent à faire ce réglage pour un site spécifique.

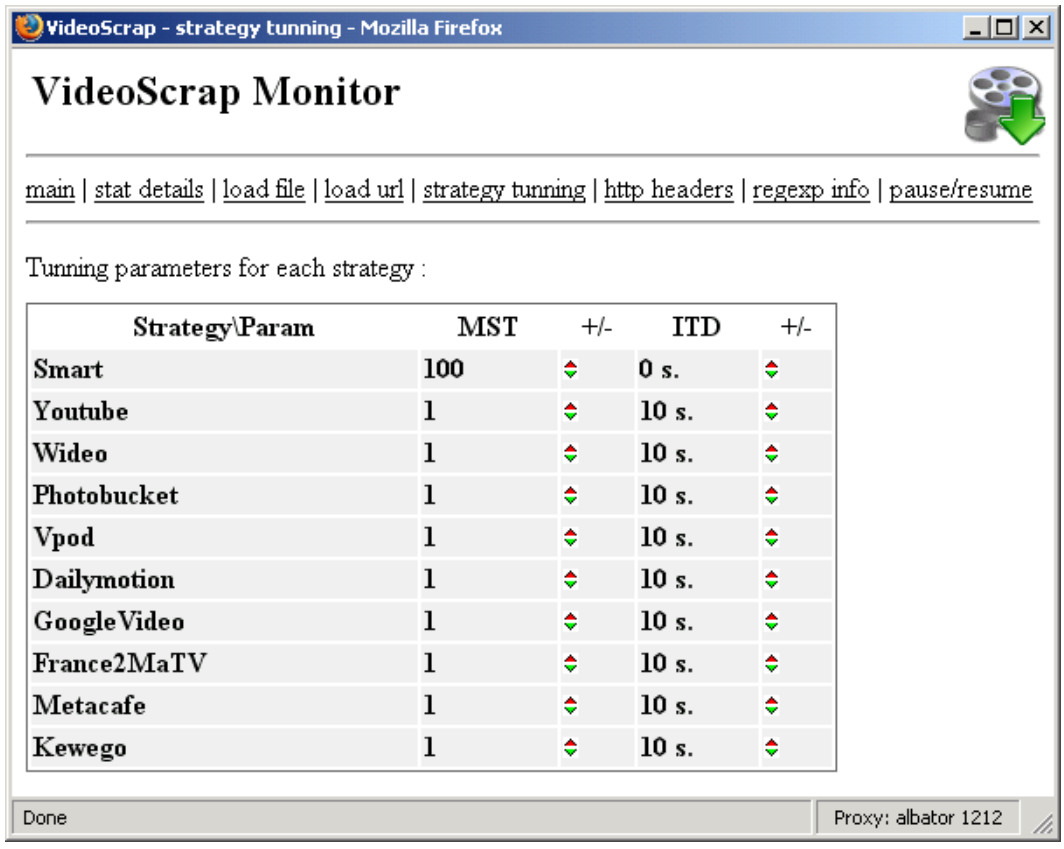


Illustration 12: réglage de la fréquence et de l'intensité de la captation

Si l'utilisateur souhaite interrompre la collecte sans l'arrêter, par exemple pour libérer temporairement des ressources processeur et réseau sur la machine exécutant la librairie, cela est possible grâce à un menu « pause/resume ». Ce menu n'arrête pas l'exécution au sens strict en raison de la nature asynchrone de la programmation de la librairie. Le comportement effectif est que les stratégies ne prendront plus de nouvelles transactions dans leur file d'attente lorsqu'une transaction en cours se terminera. Au bout d'un temps donné, plus aucune transaction ne devrait être en cours et le processus sera effectivement inactif. Cet arrêt progressif permet de réduire efficacement les ressources utilisées par la librairie.

Pour un suivi plus fin des transactions, l'utilisateur a la possibilité de cliquer sur un nombre de transactions dans le tableau de contrôle principal, la liste des transactions pour la stratégie concernée est alors affichée, classée en transactions courantes, transactions terminées et en file d'attente. Les transactions courantes peuvent être cliquées: s'affiche alors un descriptif détaillé de la transaction et de son avancement (voir illustration 13 p.28).

On peut remarquer dans le descriptif détaillé de la transaction un cadre intitulé « current transactions in entry » qui présente, s'il y en a, les transactions appartenant à la même *Entry* que la transaction vue. Ces transactions sont donc issues de la même URL de base et le fait de pouvoir passer de l'une à l'autre facilite le monitoring et la compréhension du processus.

VideoScrap - transaction details - Mozilla Firefox

VideoScrap Monitor

[main](#) | [stat details](#) | [load file](#) | [load url](#) | [strategy tuning](#) | [http headers](#) | [regexp info](#) | [pause/resume](#)

Transaction details (169468796)

go back to strategy : [Youtube](#)

current transactions in entry:

- url trace:

url	recorded ?
http://www.youtube.com/v/0O-ah8zheBM	y
http://www.youtube.com/watch_video?v=0O-ah8zheBM	y
http://www.youtube.com/jp.swf?video_id=0O-ah8zheBM&eurl=&i	n
http://www.youtube.com/get_video?video_id=0O-ah8zheBM&eur	y
http://lax-v243.lax.youtube.com/get_video?video_id=0O-ah8zheE	n

- video id : 0O-ah8zheBM
- download dir : /mnt/stockage3/videotmp/_tmp/
- already downloaded : n
- file headers :

```

HTTP/1.1 200 OK
Connection: close
Content-Type: video/flv
ETag: "-1411211802"
Last-Modified: Wed, 27 Jun 2007 13:53:00 GMT
Content-Length: 84941387

```

- message : (regexp_url_embedded_flv_redir: _get_page,_got_embedded_flv_redir)-
- thumb : none
- download progress : **9%**
- stop this transaction (will be marked as "failed") : [TODO]

Done Proxy: albator 1212

Illustration 13: affichage des détails d'une transaction en cours de téléchargement

e. Mise en oeuvre de la librairie

La mise en oeuvre des outils développés a représenté une partie importante de mon projet. L'enjeu de cette mise en oeuvre était de taille, en effet le projet de Dépôt Légal du Web est actif et des corpus définis doivent déjà être archivés régulièrement dans le cadre de l'expérimentation du système. Le contexte des campagnes présidentielles et législatives a fortement influencé l'activité des blogs et sites politiques francophones qui constituent actuellement le corpus principal archivé par le DLWeb. La forte activité et les particularités de ce corpus ont induit une densité importante de vidéos embarquées (extraits télévisuels, interviews amateurs), qu'il fallait donc être en mesure de capter afin de produire une archive complète²⁴.

Certaines vidéos stockées sur des sites comme Youtube ou Dailymotion étaient rapidement supprimées en raison de conflits de droits de diffusion concernant notamment les extraits télévisuels [7]. Cela a contraint le processus d'archivage des pages et plusieurs collectes quotidiennes ont dû être programmées afin de ne manquer aucune mise à jour des liens vers de nouvelles vidéos. En parallèle, un script spécialisé était chargé d'extraire les liens de vidéos des pages archivées et de fournir ces liens à la librairie de captation des vidéos.

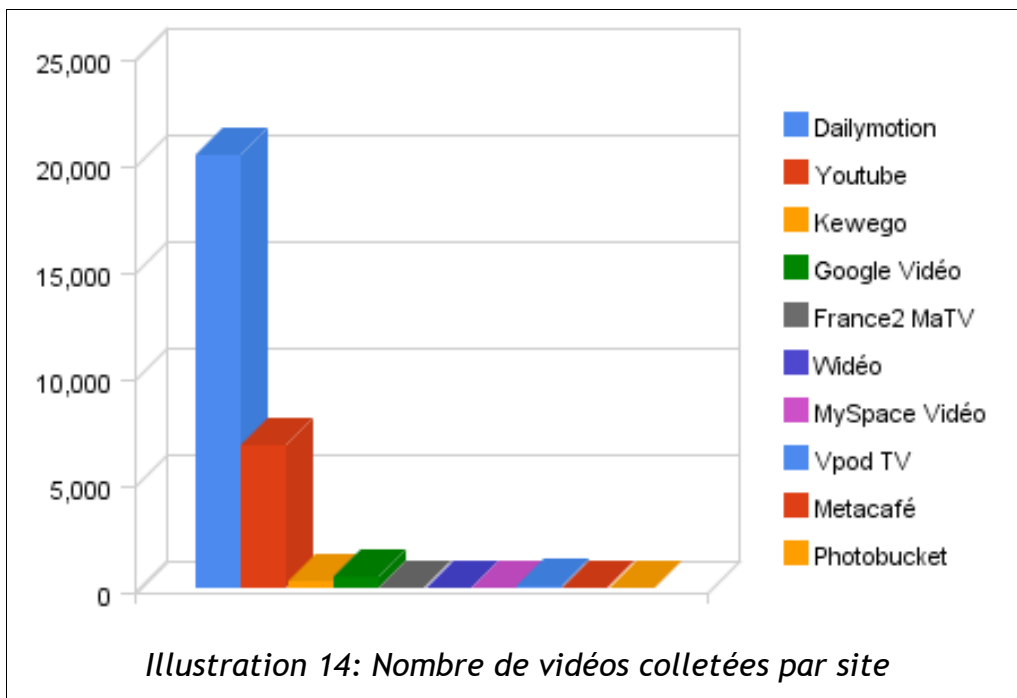
Le workflow de captation des vidéos embarquées se compose donc de trois étapes principales. En amont, les robots de captation du Web produisent des fichiers de contenus au format DAFF qui sont stockés dans des dossiers temporaires avant la phase de consolidation de l'archive. La première étape est de scanner les pages HTML dans ces fichiers à l'aide d'un script utilisant une expression régulière mise à disposition par notre librairie. Cette expression régulière est chargée de reconnaître des contenus embarqués venant des sites que la librairie est capable de traiter et d'extraire les URLs de ces contenus.

La seconde étape, une fois qu'une liste d'URLs de contenus vidéos a été produite, est de filtrer cette liste à l'aide d'un script spécialisé. Ce script a pour tâche de constituer une base de connaissance de toutes les adresses de vidéos qui ont déjà été archivées et de les supprimer de la liste qu'il reçoit en entrée. Cela évite de retraiter des adresses qui ont déjà été téléchargées et d'encombrer la file d'attente avec ces dernières. Le filtrage permet également de normaliser les URLs parfois mal formées et de supprimer un grand nombre d'URLs d'images, qui sont nombreuses à passer la première étape. Cette phase d'optimisation permet de gagner un temps précieux pendant la phase de captation des vidéos: dans les cas concrets que j'ai observés, elle permettait de réduire la taille de la liste donnée d'un facteur allant de trois à cinq.

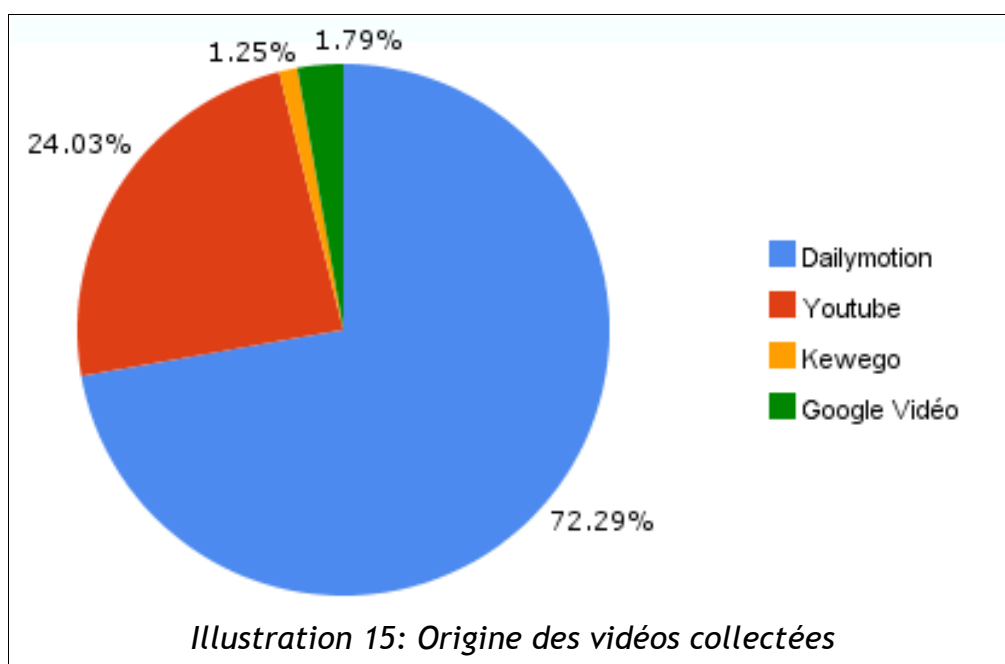
La troisième étape consiste à fournir la liste filtrée en entrée du *Harbor* décrit dans la partie précédente afin que les vidéos soient téléchargées. La phase de téléchargement est la plus longue et s'étale typiquement sur plusieurs jours pour des listes d'URLs de l'ordre de la dizaine de milliers.

Durant mon stage, plus de **28 000** vidéos ont été collectées, provenant de dix hébergeurs différents: Dailymotion, Youtube, Kewego, Google Vidéo, France2 maTV, Wideo, MySpace Vidéo, Vpod TV, Metacafé et Photobucket (voir illustration 14 p.30).

²⁴ voir à ce sujet: Annexe 5 - Collecte du traitement Web de la campagne présidentielle 2007



Parmi les vidéos collectées, on remarque une très forte présence dans les sites et blogs francophones de l'hébergeur français Dailymotion, suivi du Californien Youtube, vient ensuite Google Vidéo et enfin le français Kewego. Ces quatre hébergeurs représentent à eux seuls plus de 99% des fichiers collectés (voir illustration 15).



Avec une moyenne de **3.37** chemins archivés différents pour accéder à une même vidéo dans l'ensemble des vidéos archivées durant ce stage, un calcul rapide nous permet d'affirmer que si chaque vidéo avait été stockée à nouveau pour chacun de ses chemins, la taille totale des vidéos archivées serait de plus de **4 Teraoctets**, contre environ **500 Gigaoctets** avec notre méthode. Cela représente une économie de **700%** de l'espace de stockage et de la bande passante utilisée pour l'archivage de ces vidéos.

6.Extension Firefox : GetVideo

a.Objectifs et besoins spécifiques

Comme indiqué précédemment, les besoins relatifs au développement de l'outil de téléchargement de vidéos pour le projet Signature diffèrent de ceux concernant le projet du DLWeb. Le but ici est de permettre à un utilisateur de télécharger n'importe quelle vidéo contenue dans une page Web, sans contrainte d'efficacité ni d'enregistrement du chemin complet menant à la vidéo. L'extraction est ici semi-manuelle et non pas automatique, et l'opérateur doit pouvoir utiliser l'outil sans formation particulière.

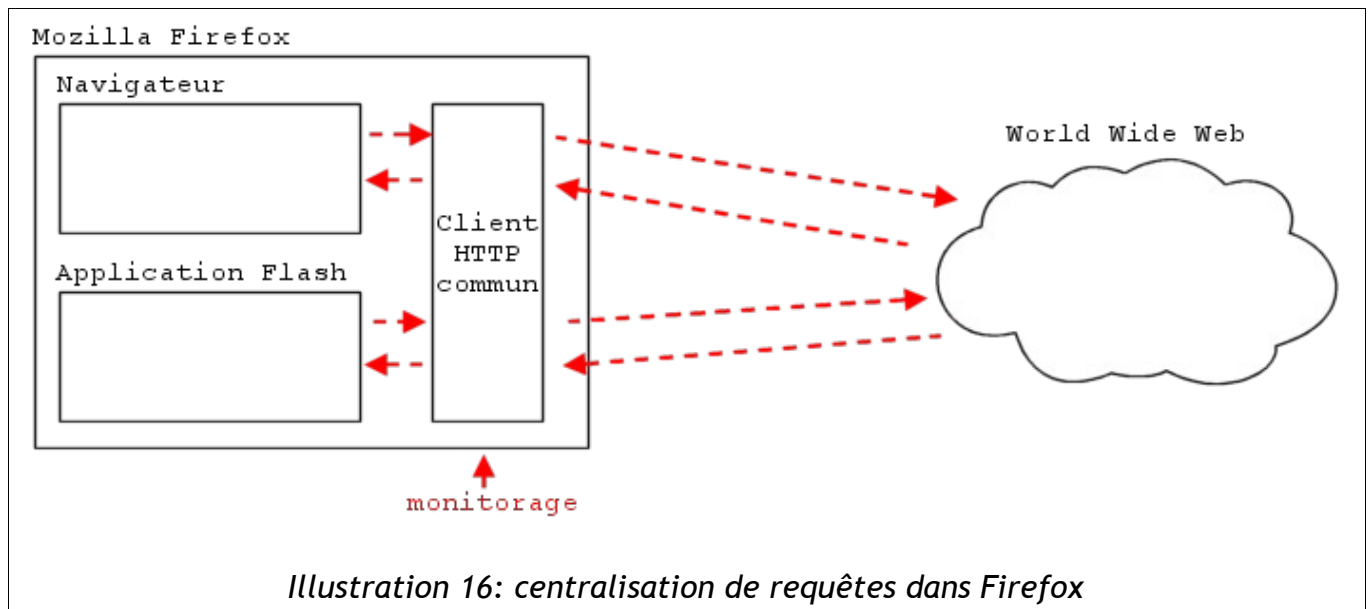
Nous avons donc la possibilité de mettre en oeuvre des techniques nous permettant une quasi-exhaustivité dans les vidéos détectées sans avoir à nous soucier de l'intégration dans un système de captation automatisé comme celui du DLWeb.

b. Étude de l'existant

Avant même de décider que le développement allait se faire sous la forme d'une extension Mozilla Firefox, il m'a fallu analyser la demande et les programmes existants y répondant éventuellement déjà. Lors de mes recherches, les seules solutions viables qui se sont présentées étaient sous la forme d'extensions Firefox. Ce résultat était pour moi prévisible car je connaissais déjà bien l'extension *Tamper Data* qui permet de monitorer toutes les requêtes qui passent par le client HTTP commun de Firefox (voir illustration 16). La connaissance de cet outil m'avait donné à penser qu'un outil similaire serait capable de répondre à la demande.

D'autres outils plus populaires, comme les extensions *VideoDownloader* et *PimpFish*, ont retenu mon attention, mais l'observation de leur code source m'a révélé que la majorité de leurs fonctionnalités était délégué à des programmes en ligne dont les résultats étaient obtenus via appel de procédures distantes. Ce fonctionnement, bien que possible, ne tirait absolument pas parti des particularités d'une extension Firefox. En effet, et comme le démontre l'extension *Tamper Data*, Firefox permet au programmeur d'une extension d'avoir accès à des données suffisamment explicites pour permettre au navigateur de reconnaître une vidéo sans l'aide extérieure d'un programme en ligne. L'appel de procédure distante me paraissait donc superflu. De plus, il s'est révélé après des tests rapides que les procédures distantes utilisées par les extensions existantes étaient des scripts PHP basés sur des heuristiques simples et fortement sujettes à l'obsolescence. En d'autres termes, elles n'étaient pas capable de détecter une vidéo sur un site « inconnu » ou « sans connaissance à priori de la structure du site ».

Mon choix s'est donc tourné vers une extension basée sur la méthode utilisée par l'extension *Tamper Data* pour monitorer les requêtes émises et les réponses reçues par Firefox.



c.Principes de développement d'extensions pour Firefox

Le développement d'extensions pour Firefox est en principe relativement simple: une extension est une archive contenant un arborescence de dossiers décrite dans un fichier *chrome.manifest*. Ce fichier décrit où se trouvent les fichiers XUL qui sont le point par lequel l'extension va se charger. Un fichier XUL est très similaire à un fichier HTML: c'est un fichier qui décrit des éléments graphiques en XML. Comme le HTML, le XUL peut contenir des liens vers des portions de JavaScript, éventuellement dans des fichiers séparés, qui vont enrichir l'interactivité des éléments graphiques décrits dans le XUL. Les éléments graphiques décrits dans le XUL doivent également spécifier leur point d'insertion²⁵ dans Firefox lui-même. En effet, l'interface de Firefox est elle-même décrite en XUL, et comme on le fait couramment en HTML, ses éléments possèdent des identifiants. Un élément XUL d'une extension Firefox doit donc spécifier l'identifiant de l'élément graphique de l'interface de Firefox dans lequel il souhaite s'insérer [8].

Ce principe, en apparence simple, cache en réalité des complications dont je ferai grâce aux lecteurs de ce document pour m'intéresser directement au code JavaScript que l'on peut attacher aux fichiers XUL. Ce code JavaScript est particulier car il peut accéder à des objets globaux de Firefox et notamment des *services*. Ces services peuvent se présenter sous la forme de *listeners* d'événements: notificateur de fin de chargement de page, notificateur de réponse à une requête; ou encore d'accès à des fonctionnalités au coeur de Firefox: lecture de paramètres utilisateurs, téléchargement d'un fichier. Grâce à ces services, le JavaScript va pouvoir interagir en profondeur avec le navigateur Firefox et avoir accès à de nombreuses données, notamment celles qui nous intéressent dans ce cas précis.

La plateforme Firefox, qui peut à la lumière de ces brèves explications, être vue comme une puissante plateforme d'applications, possède également quelques limitations qu'il est important de connaître: le fait que son développement soit principalement bénévole et libre induit une certaine instabilité d'une version à l'autre qui ne serait vraisemblablement pas de la même ampleur avec un produit de type commercial. Cette instabilité, également lié à la jeunesse du projet, a des implications très précises dans notre cas: les extensions Firefox doivent être écrites pour des versions compatibles avec le formalisme utilisé. Il faut savoir que le formalisme qui a été décrit ci-dessus possède des variantes dont la version actuelle est celle qui concerne la version 2 de Firefox. L'extension développée n'est donc pas compatible avec des version antérieures à la version 2 et ne sera vraisemblablement pas compatible avec Firefox 3 à venir [9].

²⁵ Cette insertion est appelée un *Overlay*

d.Principe et implémentation de l'extension

Le principe de l'extension est très simple: nous savons que lorsqu'on lit une vidéo via Firefox, même si cette vidéo est lue par un lecteur tierce partie (Windows Média Player, Flash, Real Player), la requête permettant de lire le fichier vidéo distant passe par le client virtuellement unique de Firefox (voir illustration 16 p.32).

Grâce à l'étude du fonctionnement de l'extension *Tamper Data*, nous connaissons les services accessibles en JavaScript qui permettent d'être prévenu par un événement de la réponse à une requête HTTP. L'événement qui est émis lors de la réception d'une réponse contient des informations qui sont les *Headers* de réponse HTTP. Ces *Headers* sont des informations d'en-tête du protocole et nous informent notamment sur le type MIME, l'URL et la taille du contenu de la réponse.

En effectuant des heuristiques générales sur ces méta-données, nous allons donner un « score » à la vidéo, qui est un entier positif. Ce score est d'autant plus grand que nous considérons élevée la probabilité que la réponse reçue corresponde à un fichier vidéo. Nous avons fixé un seuil de score au delà duquel nous considérons avoir éliminé le bruit correspondant aux faiblesses de nos heuristiques et n'avons pas perdu de résultats significatifs. Nous sélectionnons toutes les réponses ayant un score supérieur au seuil défini: les URLs considérées comme correspondant à des vidéos sont affichées et le fichier vidéo correspondant à l'adresse peut être téléchargé sur demande.

L'architecture est à l'image du principe: très simple. Nous avons une classe centrale *GetVideo* qui instancie un *listener* particulier « écoutant » les réponses reçues. À chaque réponse, un objet *Candidate* est instancié. Cet objet contient les données des *Headers* de réponses dont nous avons besoin : le code de réponse HTTP, la taille de la réponse, l'URL du contenu ainsi que son type MIME. L'objet *Candidate* contient les heuristiques de détermination du score; immédiatement après sa création, il lui est demandé de calculer le score correspondant aux méta-données qu'il contient. Si le score est supérieur au seuil, l'instance de l'objet *Candidate* est stockée dans une liste des vidéos détectées. À chaque nouvelle vidéo détectée, *GetVideo* demande à la classe *gvGUI*, chargée de l'affichage, de mettre à jour la liste affichée à partir de la liste des vidéos détectées. La classe *gvGUI* délègue l'affichage de la liste au composant *GetVideoTreeView* héritant d'une classe spécialisée dans l'affichage de listes.

La classe *gvSettings* est chargée des interactions avec la fenêtre de préférences et de configuration.

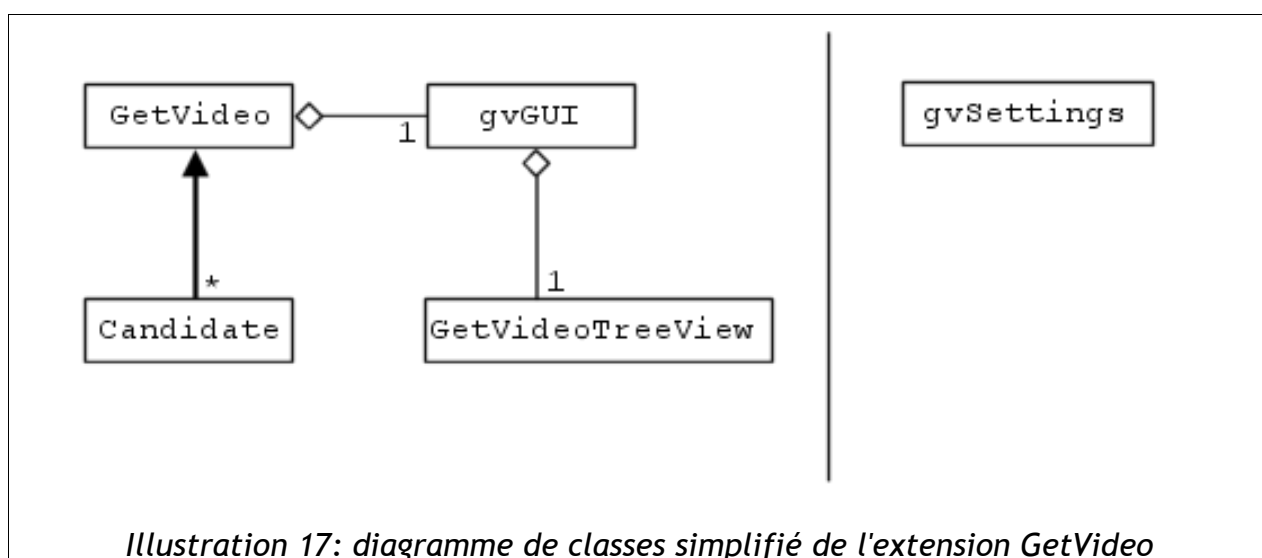


Illustration 17: diagramme de classes simplifié de l'extension *GetVideo*

Les heuristiques implémentées dans la classe *Candidate* sont des heuristiques à priori non adaptées à des vidéos en streaming. En effet, le score est proportionnel à la taille du fichier reçu, car les fichiers vidéo sont souvent de taille relativement élevée. Les fichiers de vidéos en streaming ne contiennent pas les données de la vidéo elle-même mais des données concernant le flux à lire, comme son URL et la durée de la vidéo. Pour prendre l'exemple des fichiers ASX, qui sont des fichiers de vidéo en streaming pour Windows Media Player, les fichiers ne contiennent que l'URL du flux à lire sous différents formats. Les heuristiques ont donc été modifiées pour reconnaître et donner un score élevé aux fichiers de vidéo en streaming malgré leur petite taille. Ces heuristiques se basent notamment sur le type MIME déclaré du contenu et l'extension du fichier.


Afin d'améliorer le rendement global des heuristiques, des informations spécifiques viennent leur apporter plus de précision. Ainsi, au travers du menu de configuration et de préférences, l'utilisateur de l'extension peut renseigner la liste des extensions connues pour des fichiers vidéo, ainsi que les extensions connues pour des fichiers vidéo en streaming. De plus, une liste des sites Web spécialisés dans l'hébergement de vidéos est personnalisable. Par défaut, les listes sont remplies avec les formats courants et les sites de notoriété publique.

La technique utilisée pour télécharger les vidéos diffère selon que la vidéo se présente sous la forme d'un fichier ou d'un flux. Dans le premier cas, le plus simple, il suffit d'utiliser un service de Firefox permettant d'envoyer l'adresse du fichier vidéo au gestionnaire de téléchargements intégré à Firefox et le tour est joué. Pour le streaming, qui existe à priori pour limiter ou éviter le téléchargement, un programme tierce partie doit être utilisé. Dans notre cas, nous avons choisi le programme GetAsfStream qui est capable de capter des flux MMS (Microsoft) et RTSP (Real Media). Ce logiciel a été choisi car il peut être contrôlé en ligne de commande par Firefox, ce qui permet l'automatisation de la chaîne de téléchargement. De plus, GetAsfStream est libre de droits, ce qui était un pré-requis évoqué dans le sujet de stage. Bien que l'utilisation de VLC²⁶ ait semblé plus appropriée dans un premier temps, le fait que ce dernier ne supporte pas la configuration d'un proxy HTTP ne permettait pas son utilisation sur les postes standards de l'INA, notamment ceux du service juridique à qui l'extension est destinée.

²⁶ VLC est un lecteur/transcodeur de vidéos et de flux open source populaire

e. Utilisation de l'extension Firefox

L'extension Firefox a été conçue pour être installée et utilisée par des personnes non expertes en informatique. Ces personnes, en l'occurrence le personnel du service juridique de l'INA, devaient être assistées de la manière la plus simple possible dans leur tâche. L'interface a été développée avec un souci de simplicité et d'efficacité. Les réglages éventuels à faire sur l'extension ont été introduits pour des utilisateurs experts mais ni leur compréhension ni leur utilisation ne sont nécessaires pour une utilisation classique.

L'extension se présente sous la forme d'un fichier XPI. Lorsqu'on ouvre ce fichier avec Firefox, l'extension est automatiquement installée après redémarrage de Firefox. Une fois l'extension installée, un nouveau bouton apparaît à côté des boutons « précédent » et « suivant » du navigateur. Le bouton  permet d'afficher et de cacher une barre latérale (voir illustration 18).

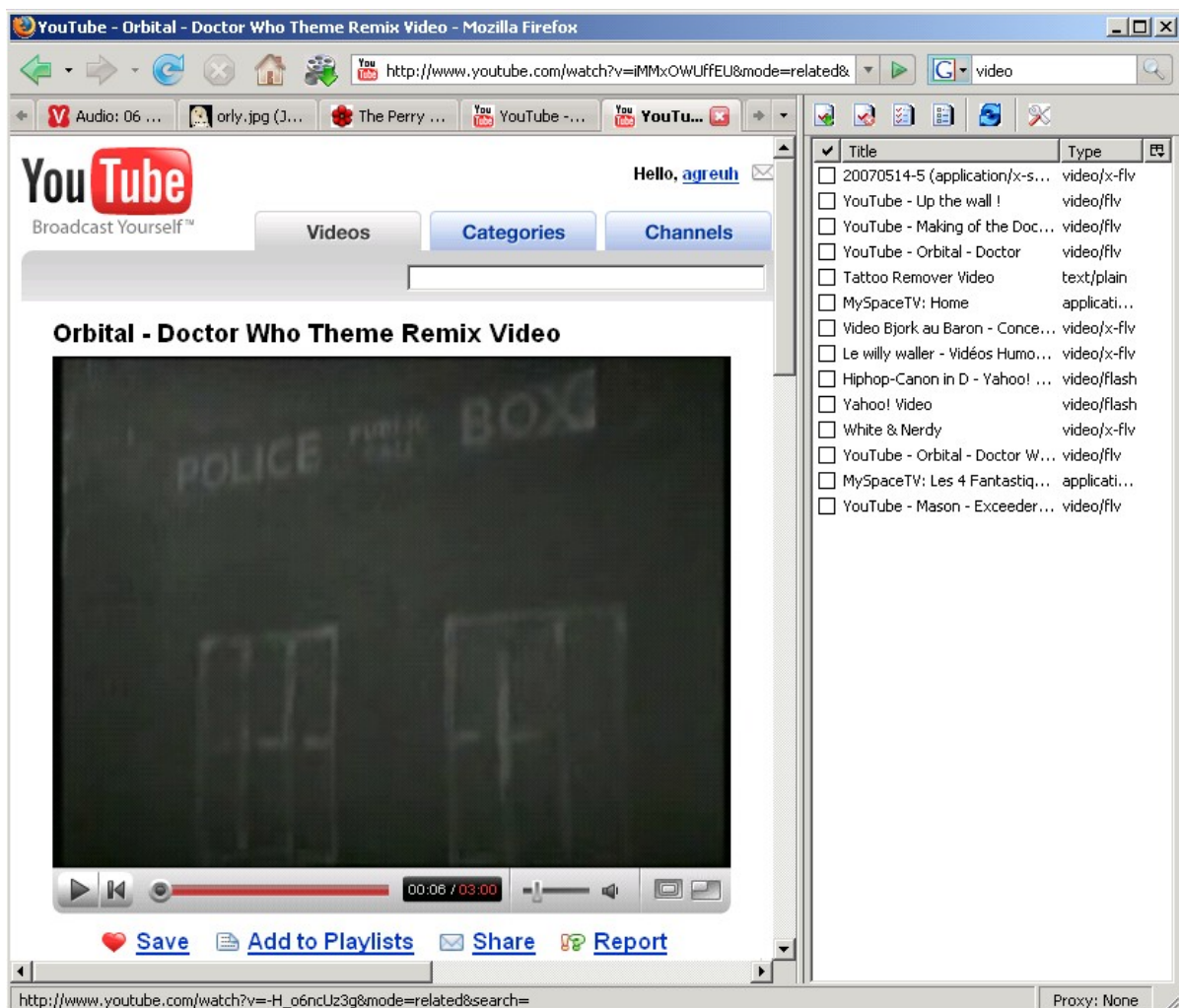


Illustration 18: interface de l'extension Firefox GetVideo

La barre latérale ajoutée à gauche de la fenêtre se compose de 6 icônes en en-tête et d'une liste vide. Lorsque la barre est ouverte, la détection de vidéos est active: lorsque l'utilisateur navigue sur le Web, toutes les vidéos détectées sont ajoutées à la liste. Les informations affichées peuvent être modifiées (voir illustration 19 p.37) mais les informations par défaut (nom de la vidéo et heure de la détection) suffisent pour une utilisation classique.

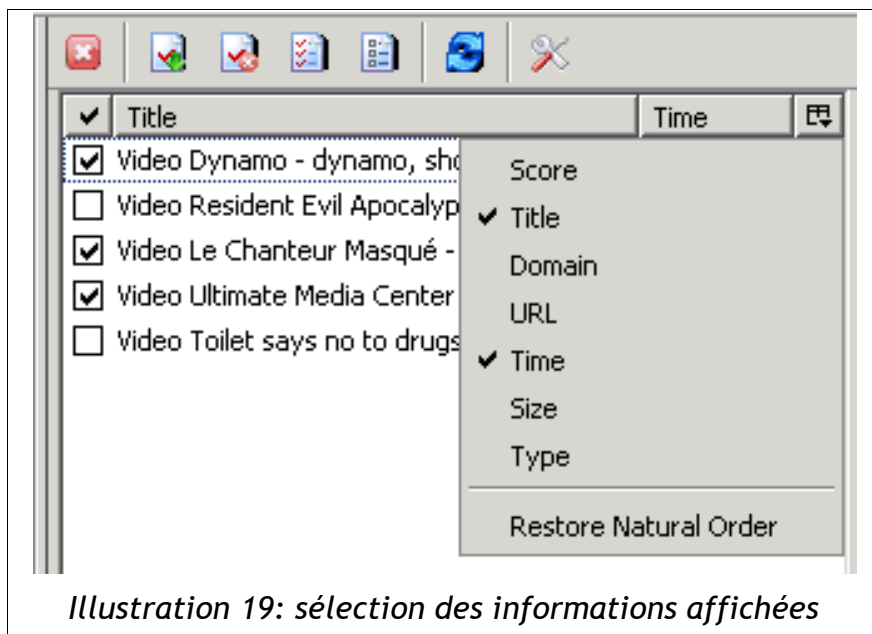







Illustration 19: sélection des informations affichées

On peut remarquer sur l'illustration 19 que la première colonne du tableau ne peut être cachée: il s'agit d'une colonne contenant des cases de sélection. Chaque vidéo peut être cochée et décochée afin de permettre à l'utilisateur d'effectuer une sélection parmi les vidéos détectées, les icônes de l'en-tête permettent ensuite d'effectuer des actions sur les vidéos sélectionnées. Ainsi, le bouton  permet de télécharger toutes les vidéos cochées et  permet de supprimer de la liste toutes les vidéos cochées. Les boutons  et  permettent de cocher et décocher toutes les vidéos respectivement. Enfin, le bouton  permet de supprimer toutes les vidéos de la liste.

Le bouton  affiche la fenêtre de configuration de l'extension (voir illustration 20).

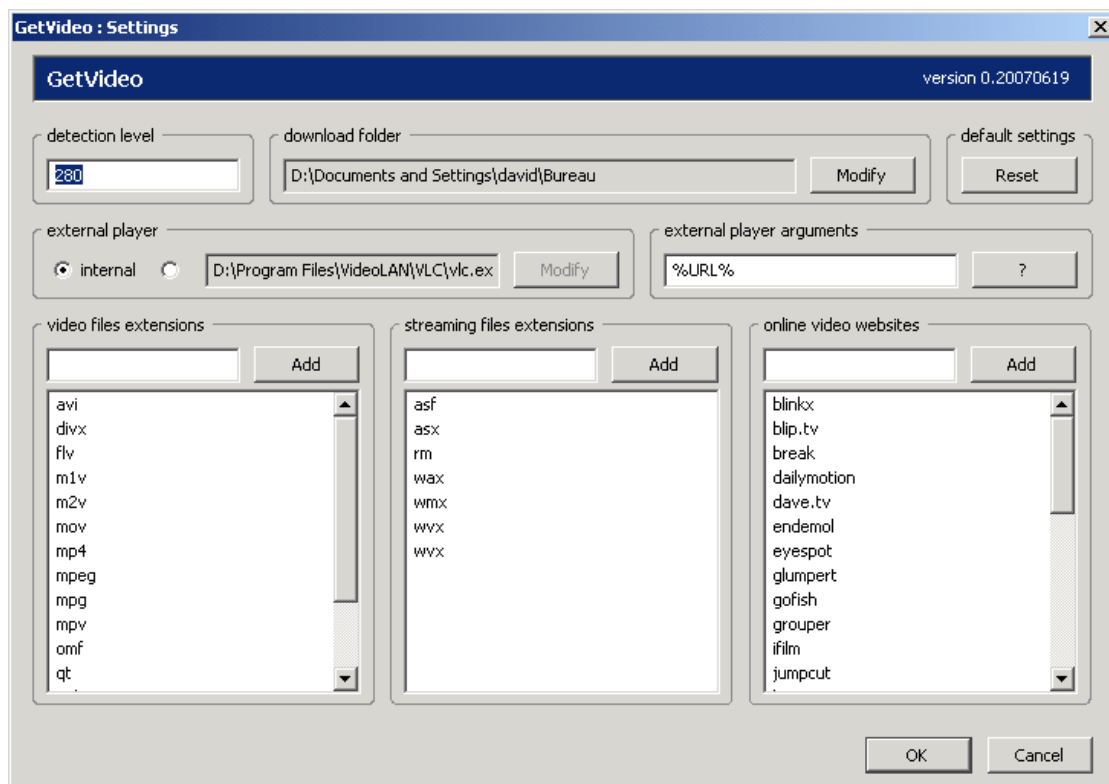


Illustration 20: configuration de l'extension GetVideo

IV.Synthèse

1.Projet de librairie Perl

Le projet de développement de la librairie Perl de collecte de vidéos pour l'archive du Web a été pour moi l'occasion de pratiquer le langage Perl de manière intensive et de parfaire ma connaissance de nombreuses technologies du Web. Durant le déroulement de ce projet, mon tuteur m'a laissé une grande indépendance en ce qui concernait les choix techniques et les décisions d'implémentations, tout en restant toujours disponible pour me donner des conseils et m'expliquer le fonctionnement de certaines librairies.

Mon travail a également été source d'interactions avec mon tuteur de stage. Ce fut le cas lorsque mon travail mettait à jour des dysfonctionnements dans les librairies que j'utilisais: il fallait alors résoudre en discutant du comportement correct à implémenter. Ce fut également le cas lors des nombreuses collectes de vidéos et de l'ajout de ces dernières dans l'archive finale.

L'outil réalisé durant ce stage est fonctionnel et permet d'archiver effectivement les vidéos provenant des 10 hébergeurs de vidéos les plus populaires du Web francophone. Les statistiques qui ont été faites sur les vidéos collectées montrent qu'une part largement majoritaire des vidéos collectées provient de deux sites (Dailymotion et Youtube) dont les stratégies de collecte ont été optimisées afin d'assurer un archivage correct. On peut donc considérer que la majorité des vidéos présentes dans l'archive sont traitables par la librairie, qui rend au jour le jour possible l'archivage d'une part représentative du Web francophone. Mon travail permet ainsi de constituer une archive fonctionnelle de qualité.

Autant du point de vue technique que du point de vue humain, ce projet s'est révélé intéressant et riche en problèmes dont les solutions étaient rarement « carrées ». Ces types de problèmes, liés à la nature même d'Internet, ont nécessité une approche pragmatique pour la résolution effective de la majorité des cas.

Ce type de prise de position, parfois difficile à prendre, a selon moi été bénéfique à mon esprit d'ingénieur.

2. Projet d'extension Firefox

Le projet de développement d'une extension Firefox dans le cadre du projet Signature a été l'occasion de découvrir le monde de Mozilla Firefox: la communauté de développeurs, les différents langages et les possibilités de fonctionnalités nouvelles. Cette découverte et l'apprentissage des langages de développement d'interface, par le biais de sites comme XULPlanet²⁷ ou MozDev²⁸, m'ont ouvert les yeux sur de nombreuses possibilités de développement que je n'aurais pas imaginées aussi facilement accessibles.

Passée la phase –quelque peu fastidieuse– d'apprentissage de l'architecture XUL et des conventions de développement d'extension Firefox, l'implémentation des fonctionnalités désirées a été relativement simple. Bien que n'ayant pas la souplesse du Perl, le JavaScript couplé à la plateforme Firefox permet d'ajouter des fonctionnalités très simplement à un navigateur Web déjà puissant.

Mon travail m'a permis d'être en rapport direct avec les problématiques du projet Signature et donc, indirectement, avec celles du service juridique de l'INA. J'ai eu l'occasion de participer aux réunions quotidiennes du projet Signature qui m'ont permis de développer mon esprit de synthèse tout en partageant avec l'équipe mes avancées et les problèmes rencontrés lors du développement. L'écoute ainsi que les conseils de Frédéric Dumas ont su cadrer mon travail de manière à obtenir au final un produit adapté à une utilisation au service juridique de l'INA, et ce malgré les contraintes de temps liées à la réalisation de deux projets en parallèle.

Au final, l'ajout des fonctionnalités pour la captation de vidéos en streaming, qui s'est fait en toute fin de stage, fait de l'extension GetVideo un outil potentiellement capable de capter n'importe quelle vidéo visible sur Internet, ce qui était effectivement l'objet de la demande.

Par son organisation et son objet, ce projet m'a permis de découvrir des méthodes de travail différentes et des technologies prometteuses.

27 <http://www.xulplanet.com>

28 <http://developer.mozilla.org>

3. Conclusion

On a vu que mes deux projets au cours de ce stage m'ont permis d'aborder le métier d'ingénieur de manières différentes. Les apports techniques, méthodologiques, organisationnels et humains que j'ai pu tirer de ces expériences correspondent à ce que je recherchais en faisant mon stage à l'INA.

J'ai appris, d'une part, à être indépendant sur un projet et à prendre des décisions techniques déterminantes mais aussi à respecter des spécifications précises et à m'adapter à un environnement technique complexe. Aussi, la présentation de mon travail lors de réunions fréquentes et la rédaction de ce rapport de stage m'ont permis de mettre à l'épreuve mes compétences communicationnelles, dont la maîtrise est très importante dans le milieu professionnel.

Cette expérience dans le milieu professionnel m'a également permis de fixer mes objectifs à moyen terme en ce qui concerne l'orientation de mes recherches d'emploi. Ainsi, mes deux stages à l'UTC se sont déroulés dans des environnements à faible contrainte économique avec un aspect recherche. C'est pour cette raison que j'ai recherché et trouvé mon futur emploi dans un domaine très différent des domaines que je connaissais jusqu'à présent, afin de multiplier mes expériences et mes compétences.

V. Bibliographie

- [1] Présentation de l'INA, <http://ina.fr/entreprise/>, juillet 2007
- [2] Présentation de l'Inathèque, <http://ina.fr/inatheque/>, juillet 2007
- [3] Projet de loi relatif au droit d'auteur et aux droits voisins dans la société de l'information (DADVSI), 21 mars 2006
- [4] Thomas Drugeon, « A technical approach for the French Web Legal Deposit », 2005
- [5] Internet a-t-il une mémoire?, Les Nouveaux Dossiers de l'Audiovisuel n°5, juin-juillet 2005
- [6] L'INA archive sites et blogs relatifs aux élections présidentielles, mai 2007
- [7] Notification des violations de copyright, http://www.youtube.com/t/dmca_policy, (Juillet 2007)
- [8] Comment développer une extension Firefox, <http://developer.mozilla.org/fr/docs/Extensions> , mars 2007
- [9] Modification de conventions d'écriture d'extensions pour Firefox 3, http://developer.mozilla.org/en/docs/Updating_extensions_for_Firefox_3, juillet 2007

VI. Annexes

1. Internet a-t-il une mémoire ?

(Introduction au dossier « Internet a-t-il une mémoire? » [5])

Et si, un jour, tous les livres s'effaçaient ? Et si, de par le monde, des centaines d'incendies ravageaient toutes nos bibliothèques, vidéothèques et autres médiathèques ? Ce qui apparaît aujourd'hui comme un pur cauchemar est pourtant une réalité dans l'univers du web où chaque jour, des millions d'informations disparaissent à jamais. Certes, vous pouvez toujours croire qu'il existe quelque part, soigneusement conservé, leur double matériel et c'est vrai pour une part des contenus accessibles sur le réseau. Mais c'est oublier qu'internet, plus qu'un simple vecteur de diffusion, est aussi le lieu d'une production inédite qui n'existe que sous la forme de données numériques lancées à travers le réseau.

Mais alors, comment arrêter ces flux numériques ? Comment les capter pour les ranger à leur tour au rayon de notre patrimoine culturel ?

Instable, infini, immatériel, le web semble pourtant bien rebelle à toute tentative d'archivage patrimonial. Mais si la tâche n'est pas aisée, il ne fait aucun doute que le plus puissant outil de la diversité culturelle ne peut rester sans mémoire, sans trace, un trou noir pour le futur.

En ce sens, l'Unesco, lors du premier sommet mondial de la société de l'information en décembre 2003, adopte une charte en faveur de la conservation du patrimoine numérique.

Ici ou là, les États prennent des initiatives, des lois apparaissent, des régimes de dépôt légal s'élaborent et reconnaissent le web comme objet patrimonial au même titre que l'écrit ou l'audiovisuel. En France, une loi doit être examinée, l'Ina et la BNF, investis d'une mission nouvelle, se préparent à relever le défi de la mémoire du web.

Jean-Michel Rodes, directeur de l'Inathèque

Geneviève Piéjut, déléguée auprès du directeur de l'Inathèque,
en charge de la coordination du projet « dépôt légal du web »

2. La mémoire du flux : Entretien avec Emmanuel Hoog

(interview extraite du dossier « Internet a-t-il une mémoire ? » [5])

La conservation du web français est d'actualité. Il s'agit d'un enjeu politique et culturel majeur. Cette mémoire doit être garantie de la façon la plus complète possible pour les chercheurs d'aujourd'hui comme pour les générations futures.

Emmanuel Hoog, président directeur-général de l'Ina, est également président de la Fédération internationale des archives de télévision (FIAT). Recueilli par Geneviève Piéjut et Philippe Raynaud.

Pourquoi conserver le web, en quoi est-il un objet de mémoire ?

Emmanuel Hoog : Le web n'est pas un objet de mémoire en soi. C'est d'abord un moyen d'expression et de communication et un lieu de création où circulent des informations, des savoirs, des données mais dont la fiabilité n'est pas toujours garantie. Malgré son caractère relativement neuf, il constitue un média original à part entière ; avec plus de 8 milliards de pages librement accessibles, 62 millions de sites recensés dans le monde en avril 2005, dont près de 360 000 sites .fr, c'est un média en progression constante.

Faut-il l'archiver ? La réponse de la France est positive car il s'agit de conserver un témoignage important pour notre pays, de ce qu'il a à dire, de ce que ses citoyens ont à exprimer. Est-ce un objet exceptionnel ou nouveau, quelle est la part de création inédite ou de publication de données préexistantes ? Le débat est ouvert et nous aurons probablement une forme de dépôt légal où la notion d'exhaustivité sera très largement disputée alors qu'historiquement, le dépôt légal était assimilé à l'exhaustivité. Nous verrons, à n'en pas douter, une évolution de cette notion de dépôt légal qui amènera la question de la sélection, du choix, du tri. Mais il est nécessaire de commencer dès maintenant, puisque la proportion de documents originaux augmente, et que, dans un futur proche, on peut raisonnablement le penser, leur nombre dépassera ce qui relève aujourd'hui de la copie. C'est vrai par exemple dans le domaine des médias où l'on voit un site comme arte-tv.com proposer des contenus exclusifs qui mériteraient d'être gardés.

Enfin, outre la nécessité de conserver le web comme source d'information pour l'avenir, ce sera aussi l'occasion d'offrir une pédagogie, une cartographie, une généalogie et une lisibilité de cet espace très difficilement représentable. Le sauvegarder, c'est aussi conserver une méthode d'accès à l'information autant que l'information elle-même.

Pourquoi une initiative publique de l'archivage du web ?

Les enjeux de mémoire sont toujours politiques puisque la politique, c'est l'organisation de la vie de la cité. Ainsi, participent-ils à la manière dont l'Histoire est perçue, dont on l'écrit et l'imagine. Il y a toujours interprétation, réécriture, valorisation, sollicitation de la mémoire ; on s'en sert comme témoin, repoussoir ou exemple. La mémoire est une notion qui appartient à la sphère publique. Dans notre tradition républicaine, sa gestion se fait sous l'autorité de la puissance publique : l'État, les collectivités locales ou des systèmes de partenariat. Elle est garante de la fiabilité de la collecte des documents, de leur conservation et de leur restitution. C'est par la disposition du dépôt légal qu'on neutralise également la question des droits relatifs à la propriété intellectuelle et artistique, en permettant d'y faire exception, et le devoir de mémoire en est une.

Cette expression démocratique est nécessaire. La BnF, le CNC, l'Ina, s'inscrivent dans cette tradition. Alors qu'il existe des initiatives privées prises dans des logiques de marché,

le service public permet d'offrir un service de qualité, en maîtrisant ses coûts, sans être soumis à une rentabilité à court terme. En France, le modèle patrimonial reste très puissant, il est une marque de la singularité de notre pays tout en constituant un atout pour l'avenir. Par exemple, les chercheurs de nombreux pays, y compris américains, qui travaillent sur les médias disposent de très peu de matériel pour leurs recherches historiques lorsqu'il n'y a pas de dépôt légal. Nous avons la chance d'avoir une recherche très vivace dans ce domaine parce que nous disposons d'un outil très puissant. C'est une force pour le futur, la capacité de notre pays à soutenir les chercheurs est un enjeu capital.

Quel sera le domaine conservé par l'Ina ?

Quand je suis arrivé à l'Ina en février 2001, de nombreuses réflexions étaient en cours sur l'archivage du web, sur l'organisation juridique, sur la méthodologie, la captation, la restitution. Dès l'arrivée de Jean-Noël Jeanneney à la BnF en 2002, nous avons eu des échanges à ce sujet et admis qu'on ne serait pas trop de deux pour être à la hauteur de cette tâche. Le projet de loi « droit d'auteur et droits voisins dans la société de l'information » étendant le périmètre du dépôt légal aux sites web prévoit d'en confier la responsabilité à l'Ina et à la BnF. Dans cette perspective, la répartition des domaines sur laquelle nous travaillons repose sur nos périmètres naturels, la cohérence étant assurée, de part et d'autre, par rapport à nos collections, nos savoir-faire, nos capacités techniques, très complémentaires, étant donné le caractère particulièrement hybride du web. Il fallait aussi que la restitution soit neutre et que les chercheurs ne soient pas pénalisés en raison de l'existence de deux institutions.

Sur ces bases, l'Ina s'intéresse au domaine de la communication audiovisuelle, c'est-à-dire à l'ensemble des sites de radio et de télévision et aux sites médias associés. Ce domaine représente, en quantité, un nombre réduit de sites - moins de 10 % du total des sites français - mais en volume, compte tenu de la présence des images et des sons, il n'en pèse pas loin de la moitié. C'est un partage intelligent qui évite les redondances et on ne peut que saluer la collaboration entre deux grands services publics pour réaliser ce grand projet.

L'archivage du web constitue une nouvelle strate de mémoire : dans quelle stratégie générale de l'Ina s'inscrit-il ?

Notre mission est de garantir que la mémoire du message audiovisuel en France soit la plus complète possible. Le dépôt légal du web rentre donc dans le cadre de nos missions car il est une nouvelle forme de mise en circulation des images et des sons, à côté du hertzien, du câble, du satellite, de la TNT. Certes, nous ne sommes pas dans un rapport de complète exhaustivité : ainsi, les radios locales de Radio France ne sont pas collectées et le dépôt légal de RFI ne se fait que sur le signal parisien. Nous gérons donc un certain nombre d'exceptions, pour des raisons d'économies, dans le cadre de notre philosophie générale. Mais globalement, nous archivons plus de 500 000 heures par an, soit une couverture très fidèle de la circulation du message audiovisuel en France. Nous devons donc être présents en matière d'archivage du web. Nous sommes dans une société où les supports de diffusion se superposent et sont de plus en plus interchangeables. Alors que nous avons une législation très liée aux supports, demain, la régulation des contenus se fera indépendamment des supports. Il faut que notre dispositif technique accompagne ces changements. L'Ina, en ce qui concerne les images et les sons sur internet, se situe dans la cohérence historique de la loi de 1974 et de la loi de 1992.

Reste la question de l'accès à ces ressources. L'accès des chercheurs à l'information est l'un des enjeux du dépôt légal du web. Aujourd'hui, nous avons un dépôt légal de radio et télévision à caractère national - demain, du web - avec une restitution uniquement

parisienne et locale. Il y a donc une grande divergence entre l'ère géographique couverte, l'étendue des contenus archivés et la mise à disposition actuellement offerte, à un niveau qui demeure trop faible, alors que l'intérêt pour ces recherches augmente : formation à l'image comme outil pédagogique, à l'audiovisuel, travaux universitaires. Néanmoins, l'accès à ces informations, sous contrôle pédagogique, est encore réduit. Il a lieu dans des emprises spécifiques, à la BnF et à l'Inathèque. Le nombre croissant des chercheurs en ce domaine m'amène à poser cette question : le dépôt légal du web pourra-t-il être accessible en région et dans quelles conditions ? Peut-on imaginer que le dépôt légal de l'internet ne soit pas accessible aux chercheurs du monde entier sur internet ? Certes, la problématique des droits existe, mais la transposition de la directive européenne consacre bien le dépôt légal comme une exception à cette règle et la gestion de cette exception devrait être moins restrictive si l'on veut que le savoir se diffuse le plus largement possible.

3.Cadre légal du Dépôt Légal du Web : la DADVSI

(Extrait du projet de loi, adopté par l'assemblée nationale en première lecture, après déclaration d'urgence, relatif au droit d'auteur et aux droits voisins dans la société de l'information; le 21 mars 2006) [3].

TITRE IV DÉPÔT LÉGAL Article 21

Le dernier alinéa de l'article L.131-2 du code du patrimoine est remplacé par deux alinéas ainsi rédigés:
«Les logiciels et les bases de données sont soumis à l'obligation de dépôt légal dès lors qu'ils sont mis à disposition d'un public par la diffusion d'un support matériel, quelle que soit la nature de ce support.
«Sont également soumis au dépôt légal les signes, signaux, écrits, images, sons ou messages de toute nature faisant l'objet d'une communication au public par voie électronique.»

Article 22

L'article L.131-1 du même code est complété par un alinéa ainsi rédigé:
«Les organismes dépositaires doivent se conformer à la législation sur la propriété intellectuelle sous réserve des dispositions particulières prévues par le présent titre.»

Article 23

I.-L'article L.132-2 du même code est ainsi modifié:

1°Le quatrième alinéa c est ainsi rédigé:

«c)Celles qui éditent, produisent ou importent des logiciels ou des bases de données;»

2°Avant le dernier alinéa, il est inséré un i ainsi rédigé:

«i)Celles qui éditent ou produisent en vue de la communication au public par voie électronique, au sens du deuxième alinéa de l'article 2 de la loi n°86-1067 du 30septembre 1986 relative à la liberté de communication, des signes, signaux, écrits, images, sons ou messages de toute nature.»

II.-Après l'article L.132-2 du même code, il est inséré un article L.132-2-1 ainsi rédigé:

« Art.L.132-2-1.-Les organismes dépositaires mentionnés à l'article L.132-3 procèdent, conformément aux objectifs définis à l'article L.131-1, auprès des personnes mentionnées au i de l'article L.132-2, à la collecte des signes, signaux, écrits, images, sons ou messages de toute nature mis à la disposition du public ou de catégories de public.

«Ces organismes informent les personnes mentionnées au i de l'article L.132-2 des procédures de collecte qu'ils mettent en œuvre pour permettre l'accomplissement des obligations relatives au dépôt légal. Ils peuvent procéder eux-mêmes à cette collecte selon des procédures automatiques ou en déterminer les modalités en accord avec ces personnes. La mise en œuvre d'un code ou d'une restriction d'accès par ces personnes ne peut faire obstacle à la collecte par les organismes dépositaires précités.

«Les organismes chargés de la gestion des noms de domaine et le Conseil supérieur de l'audiovisuel sont autorisés à communiquer aux organismes dépositaires les données d'identification fournies par les personnes mentionnées au i de l'article L.132-2.

«Les conditions de sélection et de consultation des informations collectées sont fixées par décret en Conseil d'État pris après avis de la Commission nationale de l' informatique et des libertés.»

Article 24

..... Supprimé

Article 25

.....I et II.-Supprimés.....

III.-A.-L'article L.132-4 du code du patrimoine est ainsi rédigé:

«Art.L.132-4.-L'auteur ne peut interdire aux organismes dépositaires, pour l'application du présent titre:

«1°La consultation de l'œuvre sur place par des chercheurs dûment accrédités par chaque organisme dépositaire sur des postes individuels de consultation dont l'usage est exclusivement réservé à ces chercheurs;

«2°La reproduction d'une œuvre, sur tout support et par tout procédé, lorsque cette reproduction est nécessaire à la collecte, à la conservation et à la consultation sur place dans les conditions prévues au 1°.»

B.-Après l'article L.132-4 du même code, sont insérés deux articles L.132-5 et L.132-6 ainsi rédigés:

«Art.L.132-5.-L'artiste-interprète, le producteur de phonogramme ou de vidéogramme ou l'entreprise de communication audiovisuelle ne peut interdire la reproduction et la communication au public des documents mentionnés à l'article L.131-2 dans les conditions prévues à l'article L.132-4.

«Art.L.132-6.-Le producteur d'une base de données ne peut interdire l'extraction et la réutilisation par mise à disposition de la totalité ou d'une partie de la base dans les conditions prévues à l'article L.132-4.»

Article 25 bis (nouveau)

Le dernier alinéa de l'article 22 de la loi n°86-1067 du 30septembre 1986 relative à la liberté de communication est

remplacé par deux alinéas ainsi rédigés:

«Il contrôle leur utilisation.

«Le Conseil supérieur de l'audiovisuel et l'Agence nationale des fréquences prennent les mesures nécessaires pour assurer une bonne réception des signaux et concluent entre eux à cet effet les conventions nécessaires.»

Article 26

Le IV de l'article 49 de la loi n°86-1067 du 30septembre1986 précitée est ainsi rédigé:

«IV.-En application des articles L.131-2 et L.132-3 du code du patrimoine, l'institut est seul responsable de la collecte, au titre du dépôt légal, des documents sonores et audiovisuels radiodiffusés ou télédiffusés; il participe avec la Bibliothèque nationale de France à la collecte, au titre du dépôt légal, des signes, signaux, écrits, images, sons ou messages de toute nature faisant l'objet d'une communication publique en ligne. L'institut gère le dépôt légal dont il a la charge conformément aux objectifs et dans les conditions définis à l'article L.131-1 du même code.»

Article 26 bis (nouveau)

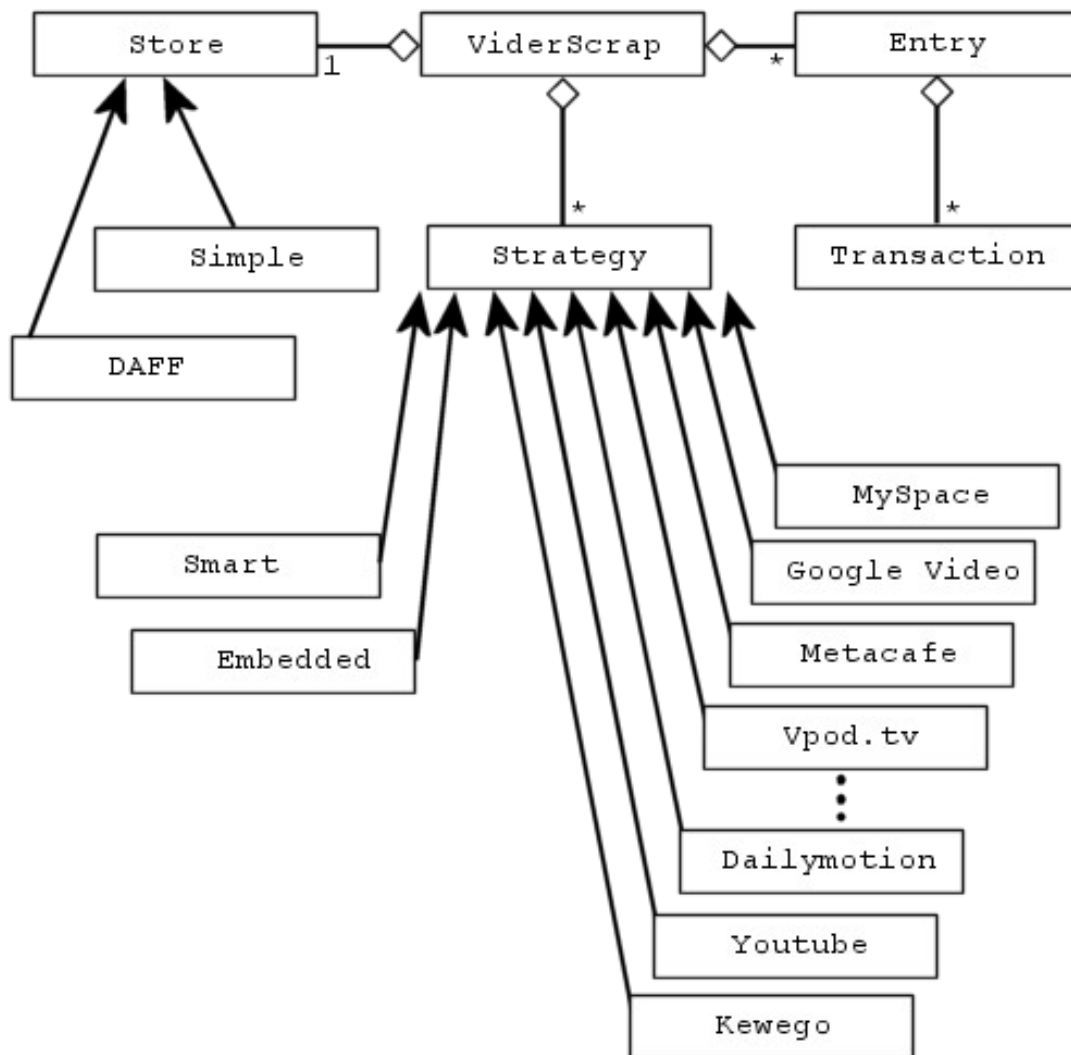
Dans les articles L.214-2 et L.311-2 du code de la propriété intellectuelle, les mots: «en France» sont remplacés par les mots: «dans un État membre de la Communauté européenne».

Article 27

L'article 2-1 du code de l'industrie cinématographique est ainsi rédigé:

«Art.2-1.-Le Centre national de la cinématographie exerce les missions qui lui sont confiées par le titre III du livreIer du code du patrimoine.»

4. Diagramme de classes simplifié de la librairie VideoScrap



5. Collecte du traitement Web de la campagne présidentielle 2007

(communiqué de presse du jeudi 10 mai 2007 : l'INA archive sites et blogs relatifs aux élections présidentielles [6])

J e u d i 1 0 m a i 2 0 0 7



L'Ina archive sites et blogs relatifs aux élections présidentielles

Dès la fin 2006 l'Ina s'est mis en situation de créer une archive, très large et très actualisée, de l'image de la campagne présidentielle sur le Web.

En s'appuyant sur les travaux de la société *Réseaux, Territoires & Géographie de l'Information* (RTGI) l'*Institut national de l'audiovisuel* a défini le périmètre du corpus des sites et blogs relatifs aux élections présidentielles.

Ce corpus intègre les sites et blogs actifs, ainsi que les sous-sites et forums politiques des principaux médias. Il est passé de 1400 sites en début de campagne à plus de 2200 depuis la mi-mars.

RTGI a mis en place le site *observatoire-presidentielle.fr*, qui présente une analyse cartographique (blogopole) et statistique (tendencologue) de ce corpus.

80 millions d'URL ont été collectées au cours de la campagne, dont plus de 15 000 vidéos (hébergées pour près de 90% d'entre elles par Dailymotion), soit près de 700Go de stockage compressé et dédoublonné, dont plus de 300Go pour les seules vidéos, dans le format de stockage DAFF développé par l'Ina.

L'opération de collecte des sites a commencé le 26 janvier. Depuis le mois d'avril, les 500 principales pages de chaque site ont été collectées jusqu'à 4 fois par jour, une collecte plus profonde étant menée tous les trois jours environ, et une collecte complète chaque mois. 220 versions de chaque site auront ainsi été captées en moyenne, à différentes profondeurs, au cours de la campagne.

Les pages d'accueil des sites ont été captées toutes les 15 minutes pendant les 2 dimanches des élections, afin de suivre l'évolution des tendances dès connaissance des premiers chiffres et sondages.