



THESIS / THÈSE

MASTER EN SCIENCES INFORMATIQUES

La télémicroscopie en cytologie hématologique

Georges, Benoît

Award date:
2002

[Link to publication](#)

General rights

Copyright and moral rights for the publications made accessible in the public portal are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognise and abide by the legal requirements associated with these rights.

- Users may download and print one copy of any publication from the public portal for the purpose of private study or research.
- You may not further distribute the material or use it for any profit-making activity or commercial gain
- You may freely distribute the URL identifying the publication in the public portal ?

Take down policy

If you believe that this document breaches copyright please contact us providing details, and we will remove access to the work immediately and investigate your claim.

Introduction

Dans le cadre de ma cinquième année à l'Institut d'Informatique des Facultés Universitaires Notre-Dame de la Paix, j'ai eu l'occasion d'effectuer un stage de cinq mois à la clinique universitaire de Mont-Godinne. Durant ce stage, j'ai travaillé sur un projet ayant pour objectif la réalisation d'un système de télémicroscopie dédié aux analyses cyto-hématologiques. Le travail consistait à développer une technique de production d'images numériques de cytologie hématologique, de qualité suffisante, et à mettre en place un réseau de consultation de ces images.

L'analyse de la faisabilité d'un tel système et la mise au point d'une station de télémicroscopie pour le laboratoire d'hématologie de Mont-Godinne avaient déjà été entamées par Monsieur Benoît Mercier. Dans son mémoire en biologie médicale [Mercier00], il présentait la possibilité de réaliser un système automatisé de production d'images cyto-hématologiques en vue du diagnostic médical. Les grandes lignes tracées par ce document ont servi de point de départ à mon travail de recherche et à la compréhension des attentes du service médical qui m'ont été adressées à l'UCL Mont-Godinne.

La télémicroscopie a émergé un peu plus tard que les autres applications des technologies numériques aux pratiques médicales telles que la téléradiologie, parce que le volume d'informations générées par les analyses microscopiques est plus important que celui d'une simple radiographie. Cependant, ces dernières années, l'évolution des technologies numériques de stockage, de traitement, de communication et de présentation des informations a rendu possible la réalisation de systèmes de télémicroscopie, permettant de rencontrer de nombreuses activités médicales en lien avec l'observation microscopique telles que la cytologie hématologique.

Ce mémoire expose les recherches effectuées et les résultats obtenus durant mon stage. Les données recueillies sont regroupées sous quatre parties : Etat de l'art, Matériel et méthode, Résultats, Discussion.

La première partie, « Etat de l'art », est consacrée à la présentation des notions de base concernant la télémicroscopie dans le cadre d'une analyse cyto-hématologique. Les concepts importants issus des expériences précédentes dans le domaine de la télémicroscopie sont analysés afin de pouvoir les appliquer par la suite au système à développer.

Cette partie débute par une présentation de la cytologie hématologique. Elle met en évidence son rôle et la manière dont s'y pratiquent les analyses afin de montrer le lien entre cette activité et l'observation microscopique.

Ensuite, à partir d'une définition de la télémicroscopie et de l'étude des systèmes existants, l'exposé se poursuit en précisant les caractéristiques devant intervenir dans un système de télémicroscopie pour que celui-ci réponde aux attentes des praticiens appelés à poser un diagnostic à partir des images fournies. Cette étude permet de distinguer deux conceptions de systèmes entre lesquels l'ensemble de ceux-ci semble se situer : l'une statique, l'autre dynamique.

L'analyse du support élémentaire et nécessaire au développement de tout système de télémicroscopie, l'image numérisée, est au centre de la problématique des systèmes télémicroscopiques. Il convient donc de connaître les concepts qui entourent l'affichage et le stockage de celle-ci .

Enfin, cette première partie se clôture par l'analyse de la structure fonctionnelle des systèmes de télémicroscopie dans laquelle on identifie les différentes opérations qu'ils doivent réaliser afin de fournir les données d'une analyse microscopique et d'en permettre la consultation. De plus, cette analyse fonctionnelle occasionne une réflexion relative à la conception d'un modèle de document multimédia adapté aux besoins des disciplines médicales.

Dans la seconde partie, « Matériel et méthode », on trouvera, d'abord, une analyse de l'infrastructure disponible à la clinique universitaire de Mont-Godinne permettant la mise en œuvre d'un système de télémicroscopie, c'est-à-dire les outils permettant l'acquisition des images microscopiques numérisées, les moyens de télécommunication nécessaires à la transmission des résultats d'une analyse microscopique et les logiciels de consultation de ces résultats. Ensuite, puisque l'objectif du projet est la réalisation d'un système de télémicroscopie dédié aux analyses cyto-hématologiques, il est indispensable de comprendre la structuration des traitements nécessaires à l'exécution d'une analyse cyto-hématologique à l'UCL Mont-Godinne. L'examen de cette organisation est donc effectué et permet de préciser le domaine d'application du système et les attentes des utilisateurs. Enfin, la méthode de conception du système est définie : le type de système à développer est choisi et sa structuration fonctionnelle est établie sur base des notions exposées dans la première partie de ce mémoire, de l'analyse de l'infrastructure disponible, de l'examen du processus d'une analyse cyto-hématologique et des attentes des utilisateurs.

La troisième partie, « Résultats », est dédiée à la conception et la réalisation du système, en respectant la méthode définie dans la partie précédente, ainsi qu'à l'exposé de son fonctionnement.

Enfin dans la quatrième partie, « Discussion », diverses lacunes du système, dans son état actuel, sont identifiées. Pour y remédier, un système plus ambitieux doit être conçu. Les bases d'un tel système sont brièvement exposées, indiquant les pistes des nouveaux développements nécessaires. La réalisation de ceux-ci doit rendre plus efficace le système, dans le but de faciliter les diagnostics que le corps médical doit poser à partir de l'analyse cyto-hématologique.

Première partie : Etat de l'art

1. La cytologie hématologique

1.1. La définition

La cytologie hématologique se définit comme le diagnostic morphologique de toute cellule de sang.

Toutes les pathologies où l'étude morphologique de cellules de sang est déterminante peuvent faire l'objet d'une analyse cyto-hématologique. La mise en œuvre de cette discipline s'intègre donc dans un processus de diagnostic général chez un patient. L'analyse cyto-hématologique est un élément qui, parmi d'autres, permet au médecin de se forger une opinion sur l'état d'un patient.

1.2. La description d'une analyse

L'analyse cyto-hématologique débute par un prélèvement de sang du patient et l'enregistrement des informations le concernant. Pour chaque prélèvement, on doit disposer des informations suivantes :

- Une identification signalétique complète et correcte du patient (nom, prénom, nom de jeune fille, sexe, date de naissance, coordonnées de facturations.)
- L'identité (lisible) et la signature du médecin prescripteur ainsi que son numéro d'INAMI et l'adresse où le rapport doit être envoyé.
- La date et le mode de prélèvement ainsi que la nature et le site du matériel prélevé (description morphologique de l'emplacement).
- Les éléments essentiels du contexte clinique. L'indication de la démarche effectuée.
- Les recherches particulières éventuelles à réaliser.

Lorsque l'enregistrement est effectué, le médecin peut passer à l'analyse proprement dite des prélèvements [Dorban99]. En cytologie hématologique, lorsqu'on compte les cellules du sang et que l'on examine leur morphologie, c'est à l'aide d'appareils (cytomètres) qui analysent les différents compartiments du sang automatiquement en effectuant des mesures cellule par cellule. Il s'agit d'appareils de routine du laboratoire d'hématologie qui permettent de voir très rapidement sur un échantillon de sang si la quantité et la qualité des éléments qu'on y trouve sont normales. Lorsque l'échantillon présente des anomalies, l'appareil les détecte. Il faut dès lors mener des examens complémentaires et le premier qui est réalisé est une analyse microscopique systématique par un des médecins du service [Dorban99]. Pour réaliser cette analyse, il faut tout d'abord étaler le sang prélevé sur une lame porte-objet. L'observation microscopique de la lame nécessite une coloration préalable qui met en évidence les structures cellulaires afin de pouvoir réaliser leur identification [Mercier00]. La lame est ensuite posée sur la platine du microscope pour être examinée par le spécialiste.

Tout examen doit faire l'objet d'un compte rendu rédigé par un cytologiste qualifié. Le diagnostic est établi au terme de l'examen des préparations microscopiques en intégrant l'ensemble des informations disponibles. Ce rapport est subdivisé en rubriques clairement distinctes :

- Les données propres au prélèvement : identité du patient (nom, prénom, nom de jeune fille, sexe et date de naissance), nom et identification du médecin prescripteur, numéro d'enregistrement dans le laboratoire, date du prélèvement, de l'enregistrement et du compte rendu et enfin la nature du prélèvement reçu.
- Les données cliniques ainsi que les antécédents du patient.

- *Le diagnostic de l'éventuel examen pré-opératoire (exemple de la présence d'une masse antérieure à l'opération chirurgicale) et le nom du praticien qui l'a effectué.*
- *La description microscopique des lésions considérées comme essentielles dans le diagnostic.*
- *Une discussion supplémentaire dans le cas d'un diagnostic différentiel. Si pour le réaliser il faut faire appel à des sources extérieures (clinique, radiologique, ...), il faut mentionner clairement leurs origines.*
- *Un bref exposé des difficultés rencontrées et éventuellement les raisons des discordances entre le diagnostic final et les examens antérieurs.*
- *Une conclusion : le diagnostic condensé reprenant les éléments essentiels de l'analyse cyto-hématologique et ses implications sur la thérapie du patient.*
- *L'identification claire et univoque des signataires [Dorban99].*

Le microscope est donc l'outil de prédilection du cyto-hématologiste. Cependant, il ne faudrait pas imaginer ce dernier comme un être solitaire qui pose ses diagnostics. Au contraire, il est indispensable pour lui de pouvoir communiquer avec les autres médecins [Dorban99]. On distingue deux types de consultation, soit entre collègues du même département, soit avec un consultant externe [DellaMea99] qui peut être localisé n'importe où.

C'est dans ce contexte que la télémicroscopie peut venir améliorer la communication et les diagnostics. Et donc la cytologie hématologique peut être considérée comme un domaine de la médecine intéressant pour la télémicroscopie.

2. La télémicroscopie

2.1. La définition

La télémicroscopie est la pratique de la visualisation et du diagnostic, à distance, d'images en provenance d'un microscope. Cette pratique nécessite la visualisation d'images numérisées sur écran vidéo plutôt qu'au travers des oculaires du microscope et également un moyen de communication entre le poste de visualisation et le microscope afin de permettre la transmission des images.

La distance qui sépare l'écran du microscope, et le support de transmission des images, ne jouent aucun rôle dans cette définition. Un cyto-hématologiste pratique la télémicroscopie aussi bien lorsqu'il visualise des images microscopiques contenues sur un CD-ROM et provenant d'un microscope situé dans une pièce voisine, que lorsqu'il visualise en temps réel des images émanant d'un microscope commandé à distance via un réseau et localisé à plusieurs centaines de kilomètres.

2.2. Les avantages et inconvénients

Si la télémicroscopie se développe via des volontés politiques, professionnelles ou industrielles, ce n'est pas par amour de la télécommunication intelligente, de l'informatique appliquée mais bien pour des raisons pragmatiques de productivité supplémentaire, de réduction des coûts de traitement ou de décloisonnement des connaissances.

Ainsi pour les cyto-hématologistes, la télémicroscopie permet de satisfaire les besoins en communication tels que : la demande d'avis, la demande d'expertise ou de diagnostic lors de la découverte d'une problématique qui dépasse le cadre du savoir du cyto-hématologue, la transmission de résultats, la formation et l'apprentissage à distance.

Ces besoins peuvent déjà être satisfaits en employant des méthodes de communication « traditionnelles », telles que : le téléphone, le courrier ou le déplacement physique des personnes. Ces techniques sont cependant moins performantes et souvent plus coûteuses. La télémicroscopie permet d'outrepasser les limites de celles-ci : lenteur de diffusion, pauvreté des supports, caractère contraignant de la localisation géographique, etc. .

Par exemple, il est fréquent qu'un spécialiste belge désire demander un avis à un spécialiste localisé à l'étranger. Dans ces circonstances , le spécialiste belge ne peut pas se déplacer pour rencontrer son homologue étranger. Il n'est également pas possible d'envoyer un courrier contenant le prélèvement, car le délai de réponse serait trop important. Le téléphone reste la solution la plus envisageable. Dans ce cas, comment rendre un avis correct sans disposer du support visuel.

La télémicroscopie peut également être envisagée dans une optique économique de réduction des coûts. Par exemple, l'automatisation d'une analyse permet de libérer un spécialiste pouvant dès lors être affecté à une autre tâche et permet en outre une diminution de la durée de l'analyse.

La télémicroscopie peut aussi jouer un rôle bénéfique dans l'archivage. En effet, les lames porte-objet peuvent être archivées pendant un certain laps de temps, afin de subir une analyse rétrospective ou être utilisées pour l'apprentissage ou dans des bases de données de cas spécifiques. Cependant elles sont fragiles, leur coloration peut s'altérer avec le temps et elles ne peuvent être reproduites.

Par contre, les images stockées sous forme digitale offrent une durée de conservation quasiment illimitée, ainsi qu'une faculté de reproduction illimitée [LGGM99], [LM01].

Cependant, malgré les avantages implicites du développement de la télémicroscopie, on observe encore une certaine réticence de la part des utilisateurs pour laquelle on peut trouver plusieurs fondements. Parmi ceux-ci, figurent :

La difficulté des systèmes à répondre à *l'hétérogénéité des besoins de chaque praticien*. La balle est ici dans le camp des développeurs de logiciels qui doivent produire un matériel le moins onéreux possible, dans le respect d'un maximum de contraintes, valable pour l'ensemble de la profession. Par exemple, les cyto-hématologistes souhaitent que la transmission des données soit rapide mais ils désirent également avoir un échantillon suffisamment représentatif (nombre assez important) d'images à haute résolution. Ces deux tendances sont, dans certains contextes techniques, incompatibles.

Le scepticisme quant à l'intérêt intrinsèque de la télémicroscopie, dans le sens où certains estiment que les gains de valeur ajoutée restent flous.

Le changement des habitudes de travail et la peur de l'équipement informatique, sans doute, dus au manque d'expérience des cyto-hématologistes au niveau de la télémicroscopie mais également dans le domaine informatique tout entier. Par exemple, un changement radical dans un système statique (VOIR 2.3, p. 9) de télémicroscopie par rapport à la microscopie est le fait que les praticiens ne disposent plus du champ infini d'observations que constituait la lame. Avec ce type de système, les praticiens doivent se limiter à une portion du champ d'observation constitué par un ensemble d'images.

D'autres problèmes ou interrogations surgissent également au niveau juridique et éthique.

Les hackers connaissent les avantages financiers dont ils peuvent bénéficier lors de la vente des données médicales à des employeurs ou des assurances qui, sur base de ces informations, refusent d'employer ou d'assurer une personne potentiellement plus encline à manquer des jours de travail ou engendrer des coûts de soins de santé. Il est donc nécessaire de garantir la confidentialité, l'authenticité et l'intégrité des données médicales. Le cryptage des données, l'authentification des transmissions et des personnes ayant accès aux données, représentent

des solutions pour prévenir les brèches dans la sécurité du système mais ne sont peut-être pas les seules.

Le manque de régulation adéquate relative à la responsabilité professionnelle doit également être comblé car il laisse sans réponse de nombreuses questions. Par exemple, la qualité des images sur lesquelles les cyto-hématologistes basent leur opinion soulève la question importante de la délimitation des responsabilités entre les fournisseurs du système et les spécialistes médicaux.

On peut également s'interroger sur la moralité d'envisager l'utilisation de la machine lors de la pratique de la télémicroscopie pour des performances qui s'intégreraient dans une logique économique, au détriment d'une conception moins technique mais plus efficace du point de vue médical. En d'autres termes, on peut se demander s'il est moral de privilégier le développement d'un système de télémicroscopie apportant des avantages économiques au détriment d'un système moins avantageux au niveau financier mais plus efficace du point de vue médical.

En somme, pour répondre à la question : faut-il développer la télémicroscopie ou non, il convient de se référer aux différents acteurs en la matière et d'être sobre et critique car les techniques modernes font l'objet d'attentions et de convictions qui ne sont pas toujours bien fondées [Nordrum99].

2.3. Les types de systèmes

Un système de télémicroscopie nécessite un microscope, une caméra digitale ou analogique montée sur celui-ci, un ordinateur capable d'acquérir les images en provenance de la caméra (donc muni d'une carte de digitalisation si celle-ci est analogique), un lien de télécommunication entre l'ordinateur d'acquisition et les stations de visualisation. Dans certaines configurations du système, l'ordinateur d'acquisition et une station de visualisation peuvent se confondre.

Certains types de systèmes nécessitent également des mécanismes hardware tels qu'une platine motorisée afin de permettre le contrôle du microscope à partir du software. C'est le cas lorsque l'on désire automatiser l'acquisition des images ou lorsque l'on veut manipuler le microscope en mode distant.

La télémicroscopie était divisée en deux types de systèmes : statique ou dynamique.

Un système statique est un système dans lequel un ensemble d'images sélectionnées, numérisées et acquises sont stockées afin de pouvoir être visualisées à posteriori. Le principe de fonctionnement de base d'un système statique est de sélectionner et d'acquérir un échantillon d'images représentatif du contenu de la lame afin de l'envoyer au consultant distant qui les visualisera lorsqu'il en aura l'opportunité.

Un système dynamique est un système dans lequel les images acquises par l'ordinateur sont immédiatement transmises à la station de visualisation, permettant ainsi une visualisation quasi en temps réel. Le principe de fonctionnement de base d'un système dynamique est de pouvoir contrôler à distance les fonctionnalités du microscope, telles que le déplacement de la platine, et de voir en temps réel, sur un moniteur d'ordinateur distant, les images susceptibles d'être observées à travers les oculaires du microscope [LGGM99].

Mais cette division est obsolète. La classification actuelle s'effectue de manière continue entre deux pôles, l'un représenté par le système statique (store-and-forward) et l'autre par le système dynamique (real time), entre lesquels on retrouve de nombreux systèmes hybrides.

Par exemple, on peut avoir un système hybride qui consiste à enregistrer un ensemble d'images conservant des relations entre elles ; telles qu'une image à fort grossissement

associée à une image à faible grossissement. Ensuite le système permet de visualiser à posteriori ces images en conservant l'aspect dynamique du changement d'objectif.

Un autre exemple de système hybride est le système qui consiste à sélectionner, acquérir et stocker sous forme d'une seule image numérique à haute résolution, l'ensemble de la surface de la lame à observer afin de pouvoir, grâce à un logiciel, reproduire sur cette image, l'utilisation de l'ensemble des fonctionnalités d'un microscope conventionnel.

Par contre, pour chaque système développé, il est recommandé, voir nécessaire, d'effectuer des tests d'efficacité, de fiabilité au niveau économique, technique et médical. Entre autre, il faut s'assurer de l'efficacité des diagnostics réalisés à l'aide d'un système de télémicroscopie par rapport aux diagnostics réalisés de manière conventionnelle. En cas d'analyse automatique, il faut garantir une durée limitée et raisonnable de l'analyse. Il faut également évaluer les ressources humaines, péuniaires et matérielles nécessaires à la mise en œuvre. Cette démarche critique est nécessaire afin d'évaluer la pertinence de l'utilisation du système.

2.4. Le support élémentaire : l'image numérisée

La numérisation des images est essentielle dans les systèmes de télémicroscopie puisque, en effet, pour pouvoir être traitées par les ordinateurs et être visualisées sur écran vidéo, les images doivent revêtir une forme numérique.

La numérisation d'une image est la conversion de celle-ci de son état analogique (distribution continue d'intensités lumineuses) en une image numérique représentée par une matrice bidimensionnelle de valeurs numériques déterminant les intensités lumineuses aux diverses coordonnées de l'image [Kaddour99].

2.4.1. L'affichage d'une image numérisée

Une image numérisée est affichée sous forme d'une image dont la surface est divisée en éléments de tailles fixes appelés pixels (*figure 1*), ayant chacun comme caractéristique un niveau de gris ou une couleur prélevé à l'emplacement correspondant dans l'image réelle ou calculé à partir d'une description stockée.

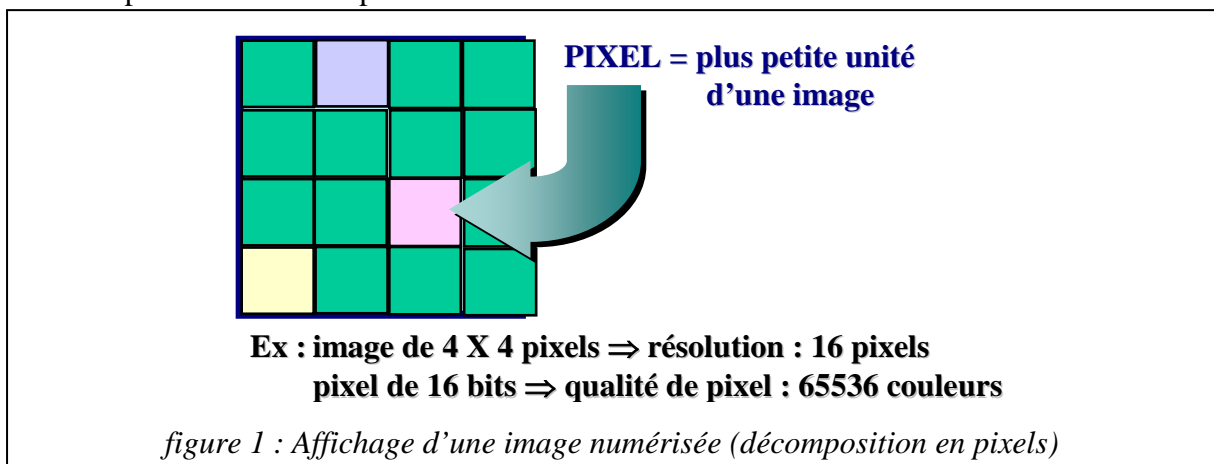


figure 1 : Affichage d'une image numérisée (décomposition en pixels)

La résolution d'une image est définie comme le nombre total de pixels qui constituent l'image par rapport à sa surface réelle, c'est-à-dire le produit de sa hauteur par sa largeur, étant donné que sa largeur est le nombre de colonnes de pixels et que sa hauteur est le nombre de lignes de pixels.

Un pixel est donc la plus petite unité d'une image numérisée pouvant correspondre à une nuance de gris ou une couleur indépendamment des autres unités de l'image. La qualité du pixel est déterminée par le nombre de couleurs différentes qu'il peut représenter. La

représentation d'une couleur par un pixel est réalisée à l'aide d'un nombre qui, pour des raisons de commodité d'adressage, est codé sur une certaine quantité de bits. Plus la quantité de bits alloués à un pixel est importante, plus le pixel peut prendre de valeurs numériques différentes et donc plus le pixel peut représenter des niveaux de gris ou de couleurs différents (figure 2).

Par exemple :		
Nombre de bits par pixel	Valeurs numériques	Nombre de couleurs ou de niveaux de gris
8	0 à 255 (2^8-1)	256
16	0 à 65535 ($2^{16}-1$)	65536
24	0 à 16777215 ($2^{24}-1$)	16777216

figure 2 : Qualité de pixel et nombre de couleurs ou niveaux de gris

2.4.2. Le stockage d'une image numérisée

En plus d'être affichée, une image numérisée doit également pouvoir être stockée sur un support physique. Le stockage des données numériques d'une image s'effectue toujours grâce à un fichier respectant un format plus ou moins standardisé. Les formats les plus courants sont JPEG, GIF, TIFF et DICOM qui est très utilisé dans le monde médical. Certains formats peuvent regrouper l'information numérique de plusieurs images dans un seul fichier, tandis que d'autres formats sont limités à une seule image par fichier. De plus, les formats de fichiers sont caractérisés, entre autres, par le mode de stockage, par le type d'image, par le modèle de représentation des couleurs et par les types de compression qu'ils supportent.

Bien que la structuration des données numériques dans un fichier diffère selon le format du fichier utilisé, un grand nombre d'entre eux sont généralement scindés en deux grandes parties. Le premier fragment du fichier est une en-tête qui contient les informations nécessaires pour interpréter les valeurs numériques représentant les points de l'image à afficher. Parmi ces informations, on retrouve le nombre de colonnes, le nombre de lignes, la compression utilisée, l'interprétation photométrique, le nombre de composantes par pixel, le nombre de bits par composante, une palette de couleurs (en cas de représentation en couleurs indexées) et le degré de compression. La seconde partie est constituée des valeurs numériques des pixels disposées en conformité avec les informations de la première partie, de manière à pouvoir être interprétées correctement.

Le mode de stockage

Une image stockée appartient, selon le mode de stockage employé, à deux grandes familles: bitmap ou vectorielle. Alors qu'une image bitmap est mémorisée à l'aide d'une matrice de valeurs numériques représentant les pixels de l'image, une image vectorielle est à l'inverse orientée vecteur. Cela signifie qu'au lieu de mémoriser une mosaïque de points élémentaires, on stocke la succession d'opérations conduisant au tracé de l'image.

Le type d'image et le modèle de représentation des couleurs

Parmi les images bitmap, on peut également distinguer les images à niveaux de gris et les images couleurs.

Dans les images à niveaux de gris, les pixels peuvent prendre des valeurs allant du noir au blanc en passant par un nombre fini de niveaux intermédiaires. Le nombre de niveaux de gris dépend du nombre de bits utilisés pour mémoriser un pixel. Si on utilise un seul bit par pixel,

les pixels de l'image sont soit noirs ou blancs. Tandis que si on utilise un octet par pixel, l'image peut profiter de 255 niveaux de gris différents.

Dans les images en couleur, les pixels ne sont plus monochromes mais peuvent représenter une série de couleurs. Cependant, pour associer les valeurs numériques stockées aux couleurs qu'elles représentent, il faut choisir un modèle de représentation.

Un premier modèle est celui de la représentation en couleurs réelles (RGB). Il consiste à utiliser vingt-quatre bits pour chaque point de l'image dont huit sont employés pour décrire la composante rouge, huit pour le vert et huit pour le bleu. Dès lors, pour déterminer la couleur d'un pixel, il suffit de combiner les valeurs de ses composantes rouges, vertes et bleues. Il est ainsi possible de représenter 16777216 couleurs différentes.

Afin de diminuer la charge de travail nécessaire pour manipuler des images en vingt-quatre bits, on peut utiliser un deuxième modèle, le modèle de la représentation en couleurs indexées. Le principe consiste à déterminer le nombre de couleurs différentes utilisées dans l'image, puis créer une table de ces couleurs en attribuant à chacune une valeur numérique correspondant à sa position dans la table. La table, appelée palette, comporte également la description de chacune des couleurs, sur vingt-quatre bits [Kaddour99]. Dès lors, pour déterminer la couleur d'un pixel, il suffit de rechercher dans la table, à la position indiquée par la valeur du pixel, la description de la couleur sur vingt-quatre bits.

D'autres modèles existent mais ne sont pas détaillés ici.

La compression

La quantité de mémoire nécessaire pour le stockage des données d'une image sans utilisation de méthode de compression peut être déterminée facilement en multipliant le nombre de pixels par le nombre de bits attribués à chaque pixel. Bien entendu, il ne faut pas négliger la place requise pour les informations, associées au format de fichier, qui emballent les données propres aux pixels de l'image. L'ensemble de ces éléments peut parfois atteindre un volume considérable. Dans ce cas, il est souhaitable d'utiliser des méthodes de compression afin de réduire ce volume.

La plupart des méthodes de compression visent à enlever la redondance présente dans l'image, de manière à diminuer le nombre de bits nécessaires à sa représentation. Les méthodes de compression peuvent se regrouper en deux classes : les méthodes sans perte d'information et les méthodes avec perte d'information, chacune ayant ses avantages et ses inconvénients [Kaddour99].

La compression sans perte

Les compressions sans perte restituent l'image originale après un cycle de compression et décompression. Les méthodes sans perte ne sont efficaces que pour des images avec des plages de couleurs uniformes et des changements de couleurs brusques telles que les logos, les textes numérisés et les dessins créés par ordinateur. Quelques méthodes de compression sans perte sont, d'ailleurs, exposées à l'annexe 1 (VOIR p. 78).

La compression avec perte

Les compressions avec perte éliminent certaines informations qui ne sont pas indispensables pour l'appréciation visuelle des images par un œil humain. Les pertes dues à la compression se traduisent par des artefacts tels que du flou sur les transitions ou des distorsions géométriques ou des réductions des couleurs. Les méthodes avec perte sont adéquates pour les images avec peu de plages de couleurs uniformes et avec des changements de couleurs fluides

telles que les photographies numérisées. Les méthodes avec perte permettent généralement de contrôler le taux de compression, en ajustant soit un facteur de qualité, soit un facteur de compression. Mais il ne faut pas perdre de vue que plus le taux de compression est élevé, plus le niveau de perte est important. La méthode de compression avec perte, JPEG est présentée à l'annexe 2 (VOIR p. 80).

2.5. La structure fonctionnelle d'un système

Le traitement complet d'une analyse microscopique grâce à un système de télémicroscopie, qu'il soit statique ou dynamique, inclut généralement la sélection et l'acquisition des images, le stockage dans un format approprié, l'intégration des informations cliniques sous forme de documents, la transmission, l'édition, la visualisation et l'interprétation des documents [DellaMea99].

2.5.1. La sélection des images

En télémicroscopie, la sélection des images consiste à déterminer un ensemble d'images pouvant être observées à partir d'une lame porte-objet, grâce au microscope et ultérieurement acquises.

On distingue trois modes de sélection des images.

Le premier mode de sélection est celui dans lequel un cyto-hématologiste passif au niveau de l'interprétation sélectionne un ensemble d'images qu'il va proposer à un second cyto-hématologiste pour que ce dernier les commente.

Le second mode de sélection est basé sur le contrôle à distance du microscope permettant au cyto-hématologiste consultant de sélectionner lui-même les images. Cependant, ce mode de sélection n'est possible que si le système conserve un aspect dynamique suffisant.

Le troisième mode de sélection consiste en une sélection automatique grâce à un logiciel des images en fonction de paramètres dont les valeurs doivent être déterminées par un cyto-hématologiste.

Le premier et le dernier mode de sélection des images peuvent mener potentiellement le consultant à des interprétations erronées car la sélection est un acte subjectif [DellaMea99] et ces modes de sélection ne laissent aucune opportunité d'interaction de la part du consultant dans la sélection des images, ni de recueil d'images supplémentaires. D'où l'intérêt du deuxième mode de sélection mis en œuvre dans un système dynamique (real-time) de visualisation.

Mais les systèmes statiques (store-forward) s'avèrent toutefois toujours nécessaires pour permettre la consultation asynchrone d'un expert dont la disponibilité est réduite. Une solution englobant les avantages des deux types de systèmes précédents peut être envisagée comme un système enregistrant l'entièreté de l'image ou la partie critique de l'image afin de pouvoir la reproduire lors de la visualisation. Ceci permettrait de conserver l'acte subjectif de sélection, ainsi que l'aspect asynchrone de la consultation.

2.5.2. L'acquisition des images

L'acquisition d'images est l'enregistrement de celles-ci dans un espace de mémoire temporaire rendant leur affichage possible à tout instant. L'acquisition d'images est généralement effectuée à l'aide d'une caméra montée sur le microscope, qui peut être soit analogique, soit digitale.

L'utilisation d'une caméra analogique nécessite une carte de digitalisation qui doit être installée sur l'ordinateur d'acquisition, afin de convertir le signal analogique venant de la

caméra en valeurs numériques supportées par l'ordinateur. La définition atteinte avec ce type de caméra peut dépasser 600 par 600 pixels pour des images de 24-bits, acquises à un régime supérieur à 30 images par seconde. *Les caméras analogiques utilisent des dispositifs à transfert de charges ou « charge-couple device » (CCD) pour recevoir et convertir les images en courant électrique. Un dispositif à transfert de charges contient une série de cellules photosensibles capables de produire une charge électrique d'intensité variable en fonction de l'intensité lumineuse qu'elles reçoivent. La performance d'un CCD est souvent mesurée par sa définition, qui dépend du nombre de cellules qu'il possède. Les CCD sont habituellement organisés en lignes de cellules [LGGM99].* Généralement, deux types d'appareils analogiques sont disponibles, l'un est basé sur un seul dispositif (monoCCD) et l'autre sur trois dispositifs (3CCD). La plupart des caméras CCD gèrent les couleurs. Dans le cas des caméras monoCCD, les cellules du CCD sont divisées en colonnes qui, chacune en alternance, filtrent le rouge, le vert et le bleu. La qualité de ces caméras est limitée par le fait que chaque couleur primaire a seulement un tiers de la définition totale et qu'aucune d'entre elles n'est filtrée sur l'entièreté de l'image. Les caméras 3CCD résolvent ces problèmes en employant trois CCD identiques, un pour chaque couleur primaire. Mais le revers de la médaille est leur prix plus élevé, leur taille et leur poids supérieur.

Les caméras digitales produisent un signal digital et sont directement connectées à l'ordinateur. Elles fournissent des images de résolution supérieure mais possèdent un régime d'acquisition moins élevé [DellaMea99], [Mercier00], [LGGM99].

Etant donné les caractéristiques des différents types de caméras, il semble que les caméras analogiques respectent mieux les exigences des systèmes dynamiques qui nécessitent des images de taille raisonnable et une fréquence de rafraîchissement élevée. Par contre, les systèmes statiques se satisfont amplement des performances des caméras digitales.

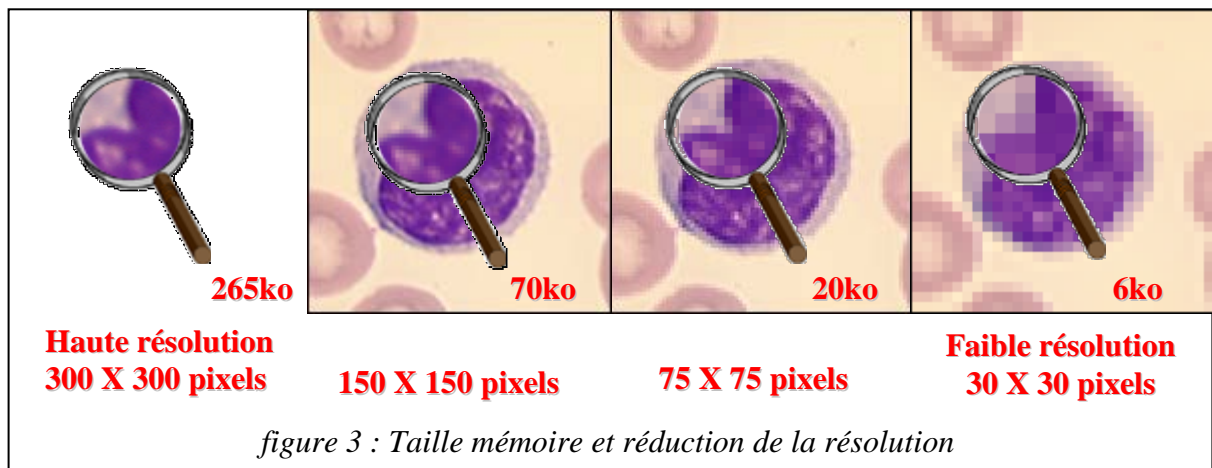
Il faut également remarquer que la qualité des images capturées dépend autant de la définition de la caméra que du grossissement de l'objectif employé. On peut donc avoir une image de qualité identique aussi bien lors d'une capture à l'aide d'une caméra à faible définition et d'un objectif à fort grossissement, qu'avec une caméra à définition élevée et un objectif dont le grossissement est faible [DellaMea99].

Enfin, l'acquisition au moyen d'une caméra offre généralement un champ de vision et une résolution limités par rapport à la vision directe au microscope [Nordrum99], [LGGM99]. Cependant, la qualité de l'image nécessaire dépend de l'objectif de l'examen microscopique et une partie seulement des informations contenues dans une image est nécessaire. Il est donc opportun de déterminer ces paramètres au lieu de se focaliser sur l'emploi d'images à haute résolution. De plus, l'efficacité d'un diagnostic n'est pas fonction linéaire de la résolution des images dont il est issu [Nordrum99]. D'autres paramètres sont à prendre en compte, tels que l'expérience du spécialiste, la qualité de la lame observée, la représentativité de l'échantillon observé, etc. .

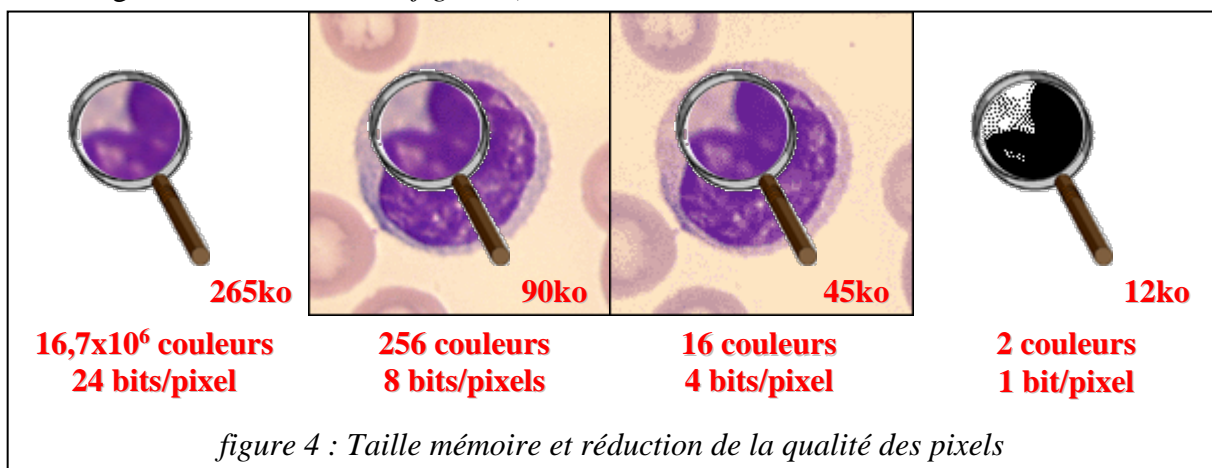
2.5.3. Le stockage des images

L'analyse d'une lame peut potentiellement générer une grande quantité de données numériques. Le stockage des images implique donc un compromis entre la quantité de mémoire nécessaire à celui-ci et la qualité des images [Mercier00]. Pour établir ce compromis, on peut faire varier différentes caractéristiques d'une image numérisée.

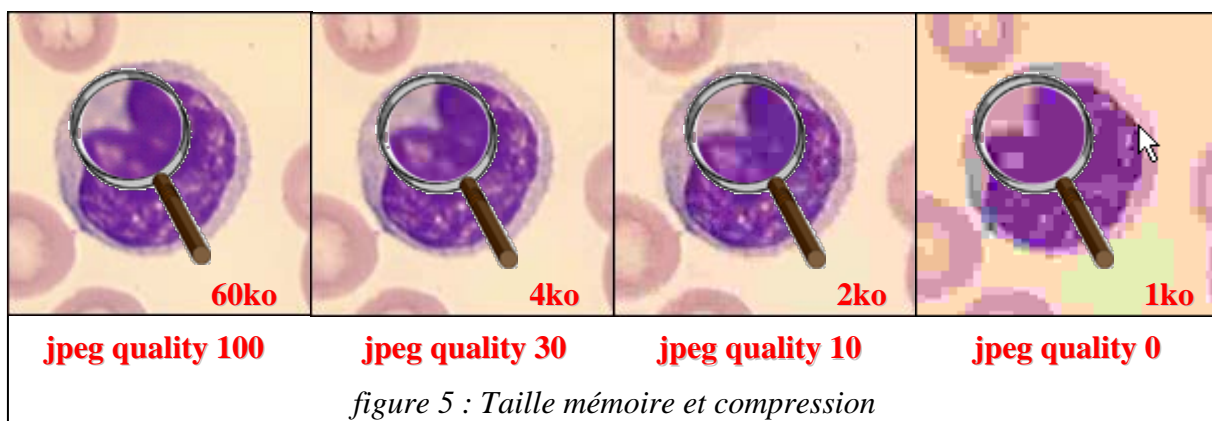
Premièrement, on peut, à partir d'une image digitalisée, réduire sa résolution, c'est-à-dire le nombre de pixels représentant l'image. Il faut savoir que toute diminution du nombre de pixels implique une diminution de la taille de mémoire prescrite au stockage de l'image mais entraîne également une baisse de qualité de cette image, appelée « effet de pixelisation » (*figure 3*).



Deuxièmement, il est possible de réduire l'importance de la mémoire requise au stockage d'une image, en réduisant la qualité des pixels, c'est-à-dire le nombre de bits alloués pour chaque pixel. Il est clair que si la qualité de chaque pixel est réduite, la qualité de l'ensemble de l'image subit le même effet (*figure 4*).



Enfin, on peut utiliser une méthode de compression afin de limiter la quantité de mémoire nécessaire au stockage d'une image. On a expliqué que les méthodes sans perte d'information accèdent à de meilleures performances quand elles sont appliquées aux images dont les pixels recouvrent une redondance importante. Or les images issues d'une analyse télémicroscopique ne possèdent pas une telle caractéristique. Au contraire, ce sont des images avec peu de plages de couleurs uniformes et avec des changements de couleurs fluides. En conséquence, seules les méthodes avec perte d'informations, peuvent aboutir à un taux de compression avantageux. Mais la perte d'informations, signifie également un perte de qualité (*figure 5*). Cependant, pour les images de cyto-hématologie, la perte de qualité reste minime même avec des taux de compression importants.



On a déterminé différentes caractéristiques dont les variations permettent d'établir un compromis entre la qualité et la place mémoire nécessaire. Il ne faut cependant pas se focaliser sur l'un des deux aspects. Ce compromis doit simplement être déterminé en fonction des besoins du spécialiste, de l'analyse qu'il veut effectuer et des ressources hardware dont il dispose. Dans certaines circonstances, on privilégiera la qualité au détriment de la taille mémoire ; tandis que dans d'autres conditions on emploiera des images comprimées à basse résolution.

2.5.4. L'intégration des données cliniques sous forme de document

La démarche médicale incrémentale, collégiale par nature, génère un flux de données. Elle nécessite donc un système d'intégration de données capable de sauvegarder, de rassembler, d'organiser et de communiquer ces données médicales. Le concept de document est particulièrement porteur de ces valeurs. De plus, les nouvelles technologies de l'information et de la communication ont mis en évidence les qualités intrinsèques de l'approche documentaire pour la communication et la présentation multimédia. Au vu de ces constatations, il est donc naturel de vouloir intégrer les images et les informations cliniques au sein d'un document multimédia.

De nombreux avantages potentiels de la télémicroscopie ne peuvent être réalisés que si ce moyen de communication est accessible universellement par un groupe restreint d'utilisateurs. L'accès universel (c'est-à-dire de n'importe quel endroit du globe), surtout dans le domaine médical, implique un ensemble d'exigences vis-à-vis du système de communication.

Tout d'abord, le système doit faire face à un ensemble très hétérogène d'applications de restitution et d'édition de documents [AHL96]. Pour résoudre cette difficulté, on peut imaginer l'élaboration d'un standard de document, sur lequel peuvent se baser les applications. Mais à l'heure actuelle, l'intégration des informations cliniques et des médias en un document unique ne fait l'objet d'aucun format standard spécifique à la médecine ou à la cytologie hématologique. Les implémentations actuelles utilisent des formats propriétaires ou le format HTML qui ne requiert qu'un Web Browser pour la visualisation [DellaMea99]. Cependant, il y a conflit entre le déploiement d'un standard et les besoins spécifiques aux différents domaines médicaux [BFWAMS99]. De plus, le système doit être compatible avec une multitude de niveaux de ressources (hardware et software) et d'environnements associés aux utilisateurs. Et enfin, il doit pouvoir fournir une présentation du document dédiée aux besoins et profile de chaque utilisateur. Or le langage *HTML* se restreint à une et une seule présentation finale, les données étant gravées dans le document [Emim02].

Pour éluder ces obstacles, une première approche propose de générer et enregistrer plusieurs versions des données selon différents formats, présentations, etc. . Mais cette solution requiert un espace mémoire trop important [AHL96].

Une seconde approche consiste à enregistrer une seule version des données et de fournir toutes les interfaces nécessaires pour convertir ces données dans les présentations ou les formats requis. Mais cette solution nécessiterait le développement d'une multitude d'interfaces [AHL96].

Une troisième approche, la plus séduisante, réside à enregistrer une seule version des données accompagnées de règles, de contraintes et associées à un ensemble de scénarios transformant les données sous différentes manifestations en fonction des différents environnements, applications et utilisateurs, suivant les règles et contraintes. Pour cela, il est nécessaire d'étudier quelles informations doivent être fournies au moment de la spécification et comment les exploiter au moment de la présentation. Il faut également remarquer que, potentiellement,

seule la manifestation du document, c'est-à-dire une partie du document multimédia, doit être envoyée à l'utilisateur [AHL96].

Les feuilles de transformation sont un moyen bien connu d'associer différentes présentations à un même ensemble de données. Et il est tout à fait possible d'en choisir une en fonction de certains paramètres, ce choix étant simplement basé sur la correspondance entre un ensemble de valeurs de paramètres et une feuille de transformation. Dès lors, leurs associations avec un document source peuvent tout à fait jouer le rôle d'une manifestation [BJK00].

Il reste donc à envisager de séparer d'une part l'information et d'autre part la manière dont elle sera rendue grâce aux feuilles de transformation. Poussant plus avant ce concept et se référant à différents travaux de recherche sur la composition multimédia, on envisage la manifestation comme une superposition de transformations apportant chacune une dimension particulière : dimension logique, dimension spatiale, dimension temporelle et dimension hypermédia (*figure 6*). En effet, depuis de nombreuses années, les documents multimédias ont fait l'objet d'études qui ont conduit à l'identification de caractéristiques attachées aux documents classées selon ces quatre dimensions. L'un des avantages importants d'une telle architecture de composition est la possibilité de réutilisation des médias pour un ensemble de transformation et, inversement, l'application d'un même groupe de transformations pour différents groupes de médias.

A partir de ces constatations, des projets tels que EMIM, visant à organiser des données en provenance de plateaux techniques d'imagerie médicale, en comptes rendus multimédias (plusieurs médias) et multimodalités (plusieurs modes d'acquisition) [Emim02] ou Madeus et *Opéra*, s'intéressant à la spécification des différentes dimensions des documents telles qu'évoquées ci-dessus, [OPERA00] se sont développés.

Dans ces types de solutions, le document multimédia est organisé autour des quatre dimensions qui doivent être représentées de manière indépendante [Emim02], [Roisin99]. Pour chacune de ces dimensions, le travail de modélisation consiste à identifier d'une part les entités de base et d'autre part leur mode de composition.

La dimension logique d'un document peut être traitée à l'aide du standard XML (Extended Markup Language) [W3C98]. Elle permet d'organiser hiérarchiquement le contenu informationnel du document. Grâce au balisage, XML permet de définir les différents liens logiques entre les éléments de base du document tels que les informations, les médias et les différentes parties de document, etc. . Par exemple, le document EMIM est défini par un fichier XML (associé à un DTD) qui caractérise son organisation logique en pages. L'organisation logique des pages en objets médias est décrite par un fichier XML (associé à un DTD) pour chaque page [Emim02].

La dimension temporelle peut être abordée à nouveau par XML. Il suffit de définir une DTD de XML spécialisée permettant de décrire les dépendances temporelles entre les éléments de base du document. On distingue trois types de modèle : les modèles basés sur des points du temps, les modèles basés sur des intervalles de temps et les modèles basés sur l'apparition d'événements [BKW00]. L'organisation temporelle des objets multimédias basée sur des points du temps, nécessite qu'à chaque objet corresponde un début et une fin. Tandis que la composition temporelle d'un document basée sur les intervalles, requiert que les objets soient caractérisés par une durée et un ensemble de dépendances entre eux [VRL00], [Roisin99].

La dimension spatiale peut être gérée grâce à XSL FO (Extensible Stylesheet Language Formatting Object) [W3C99a] qui est une DTD de XML spécialisée dans la description de dépendances spatiales entre divers éléments. Il faut remarquer que XSL (Extensible Stylesheet Language) [W3C99b] est composé de deux parties : XSL FO et XSLT (Extensible Stylesheet Language Transformation) qui est également une DTD particulière de XML, mais

cette dernière est spécialisée dans la transformation de documents XML. D'autres modèles peuvent également être étudiés. Deux modèles de positionnement spatial des éléments d'un document peuvent être distingués : le modèle basé sur le positionnement absolu et le modèle basé sur le positionnement relatif.

La dimension hypermédia qui correspond à l'ensemble des informations permettant de lier des documents ou des fragments de documents entre eux, peut être négociée avec le standard Xlink (XML Linking Language) [W3C01] qui est encore une fois une DTD particulière de XML [VRL00].

On remarque que chaque dimension peut être spécifiée par un fichier XML respectant une DTD définie. Le codage des dimensions spatiales, temporelles et hypermédiées dans des feuilles de transformation utilisant le langage de transformation spécialement créé pour la transformation de document XML, permet alors la transformation du fichier XML spécifiant la dimension logique en différentes manifestations du document multimédia dont le comportement spatial, temporel et hypermédia est décrit par les feuilles choisies pour la transformation [VRL00]. Ainsi, le processus de construction d'un document adaptable est composé de deux parties : la construction de la structure du contenu et la spécification des feuilles de transformations permettant de générer la présentation.

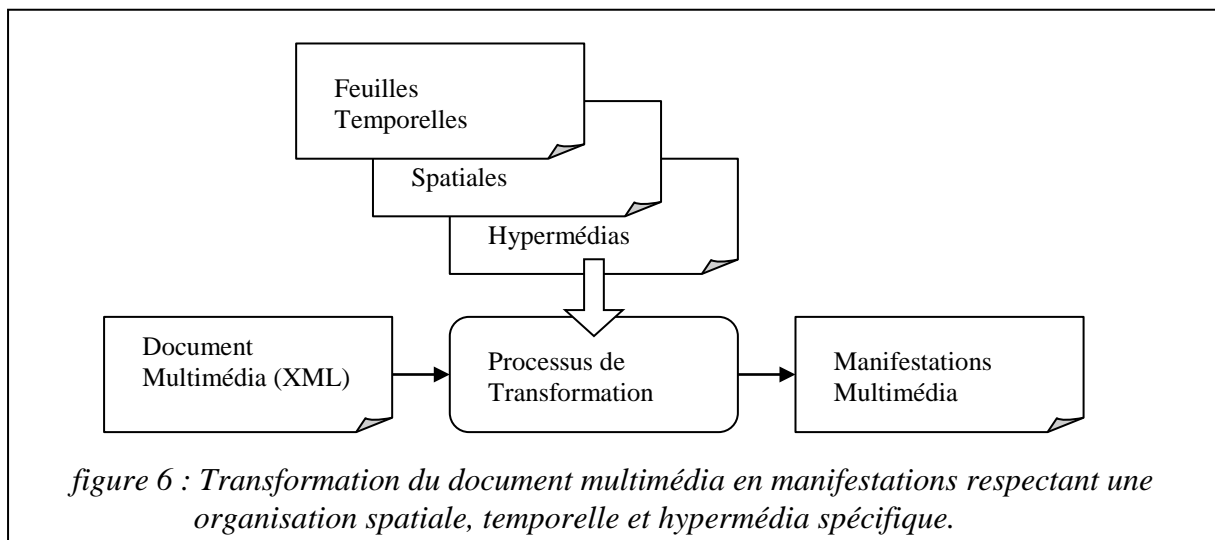


figure 6 : Transformation du document multimédia en manifestations respectant une organisation spatiale, temporelle et hypermédia spécifique.

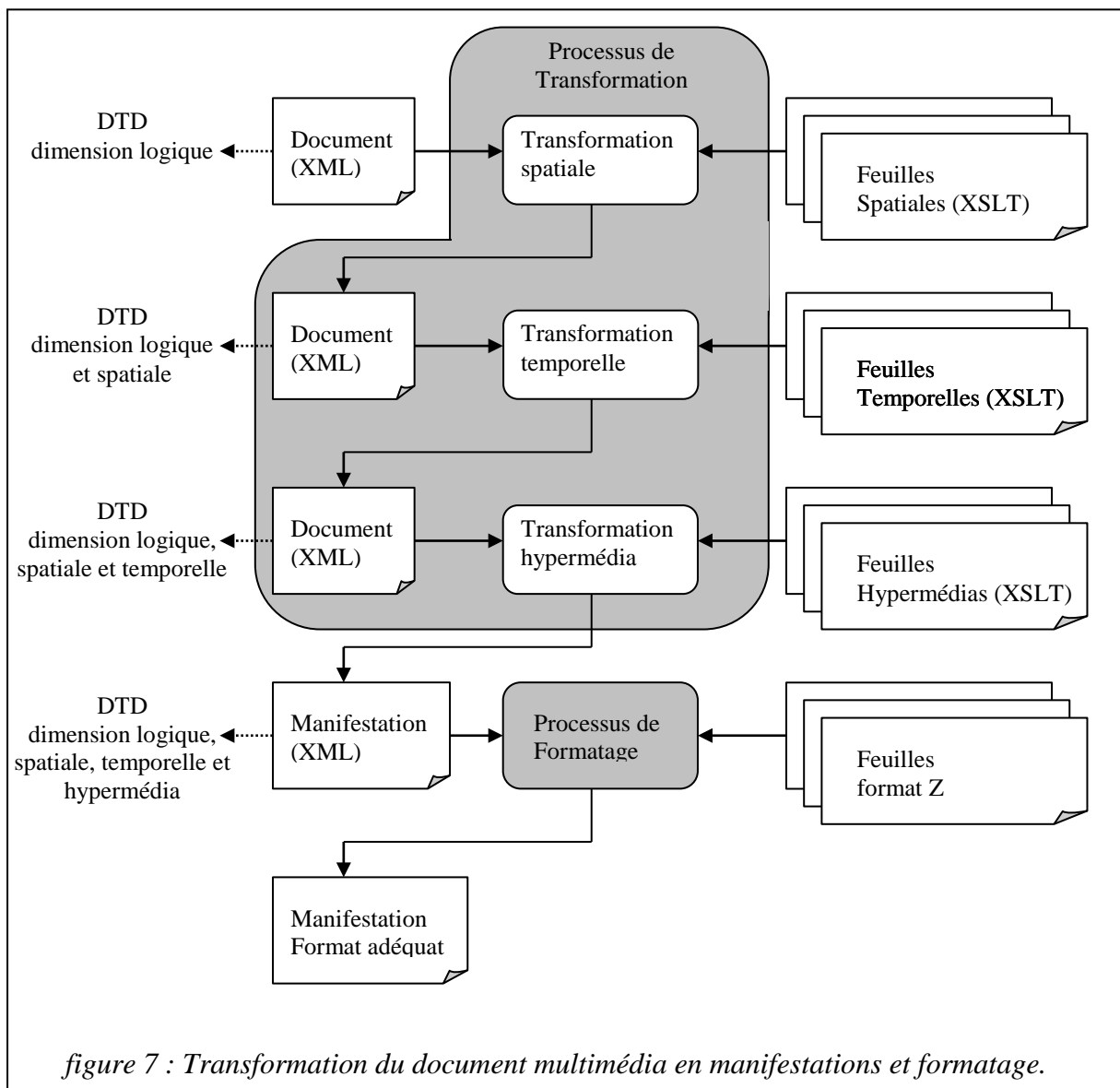
La production d'une présentation du document multimédia s'achemine en deux étapes.

Premièrement, le processus de transformation permet la création des objets multimédias, leur synchronisation et leur disposition à partir du document source et de la ou des feuilles de transformation qui spécifient les dimensions temporelle, spatiale et hypermédia. Ces feuilles de transformation consistent en des règles de transformation associées avec des patterns. Quand un pattern d'une règle s'apparente à un élément du document source, la règle correspondante est appliquée. Il est donc possible de définir une feuille de transformation pour chaque dimension du document multimédia. Si on définit un ensemble de feuilles de transformation pour chaque dimension d'un même document source, on peut alors obtenir les différentes manifestations du document [VRL00] en choisissant une feuille parmi l'ensemble pour chaque dimension. Le choix de la présentation et donc des feuilles de transformation à appliquer se fait en fonction des différents environnements, applications et utilisateurs, suivant des règles et contraintes [BJK00]. Cette étape ne résout pas le problème de portabilité entre applications. C'est pourquoi, il est nécessaire de générer une seconde étape.

Deuxièmement, le document multimédia est formaté pour être utilisable par un ensemble très hétérogène d'applications de restitution et d'édition de documents. Puisque, XSL FO et XLink sont des DTD qui définissent une syntaxe particulière de XML, qu'une DTD doit être

définie pour l'aspect temporel et que *XSLT est un langage qui a la possibilité de transformer un document conforme XML basé sur la DTD particulière en un autre, les feuilles de transformation peuvent fonctionner d'une DTD à une autre. Pour cette raison elles peuvent être utilisées en chaîne* [Apache99]. Donc, puisque le processus de transformation peut fournir un document XML, le formatage peut également employer des feuilles de transformation qui seront choisies sur les critères précédemment employés afin de fournir le document dans le format requis [VRL00].

Ce principe de séparation du processus de transformation et du formatage a pour objectif de faciliter la portabilité des documents ainsi que leur traitement par des applications variées. En effet, la mise en forme du document (c'est-à-dire la définition de la dimension spatiale, de la dimension temporelle et de la dimension hypermédia) est totalement indépendante du formatage. Cela permet de ne se préoccuper du format utilisé que lors d'une ultime transformation.



Pour conclure, on peut donc imaginer une architecture de document multimédia (*figure 7*) dans laquelle on a un document XML initial qui respecte une DTD, organisant hiérarchiquement le contenu informationnel du document (c'est-à-dire la dimension logique).

On dispose également d'une série de feuilles de transformation XSLT dédiées à la dimension spatiale parmi lesquelles on en choisit une, afin d'obtenir un document XML. Ce document XML respecte une seconde DTD traitant à la fois de la dimension logique et de la dimension spatiale.

Comme pour la dimension spatiale, on dispose d'une série de feuilles de transformation XSLT dédiées à la dimension temporelle. On choisit une d'entre elles afin d'obtenir un document XML respectant une troisième DTD « logico-spatio-temporelle ».

Enfin, on dispose d'une série de feuilles de transformation XSLT dédiées à la dimension hypermédia. On choisit une d'entre elles afin d'obtenir un document XML respectant une quatrième DTD. Ce document XML correspond à une manifestation du document initial, recouvrant les quatre dimensions principales. On peut donc considérer l'application d'un ensemble de feuilles de transformation dédiées chacune à une dimension comme l'exécution d'un scénario.

Il est aussi judicieux de remarquer que si les feuilles de transformation de chaque dimension respectent les DTD source et cible définies préalablement, c'est-à-dire si elles permettent de passer de la DTD source à la DTD cible, alors chacune des feuilles de transformation d'une dimension peut être sélectionnée avec n'importe quelle autre feuille de transformation des autres dimensions. En effet, pour chaque dimension, le document XML cible, obtenu grâce à la feuille de transformation, respecte la DTD cible de cette dernière. Or, la DTD cible de cette première feuille de transformation correspond également à la DTD source de la feuille de transformation suivante. Donc, le document XML cible peut être considéré comme document XML source de la transformation suivante, puisqu'il respecte la DTD source.

Si les feuilles de transformation de chaque dimension respectent les DTD source et cible définies préalablement, on est également assuré que les feuilles de transformation des différentes dimensions peuvent s'appliquer séquentiellement. Mais évidemment leur application doit s'effectuer dans un certain ordre, défini par la manière dont sont constituées les différentes DTD.

Le processus de transformation permet de fournir un document XML correspondant à une manifestation du document XML initial, spécifiant les quatre dimensions principales. Cependant, pour être utilisable par une application de restitution et d'édition de documents, il faut que cette manifestation revête un format acceptable pour cette application. Soit l'application est capable d'interpréter les documents XML respectant la DTD, soit une dernière transformation de formatage est nécessaire. Dans l'état actuel des choses, le choix de la deuxième solution semble obligatoire.

2.5.5. La transmission des documents

Une fois le document multimédia composé, si on utilise le type de document décrit ci-dessus, une manifestation de celui-ci doit pouvoir être transmise, via les réseaux informatiques, de manière fiable, en gérant les erreurs et les situations anormales, en considérant la sécurité mais aussi la rapidité [DellaMea99]. Il est donc propice de distinguer les types de réseaux, de prendre connaissance des performances des supports physiques de transmission, des protocoles de communication, des réseaux et des moyens de sécurisation.

Les supports physiques

Quelque soit le type de réseaux, pour transmettre un flot de bits d'informations d'une machine à une autre, il faut disposer d'un support physique de transmission. De nombreux médias peuvent être utilisés à cette fin, et chacun d'eux a sa propre spécificité en terme de bande passante, de délai, de coût, de facilité d'utilisation.

L'un des moyens les plus communs pour transférer des données d'un ordinateur à un autre consiste à les enregistrer sur un disque magnétique ou un CD-WORM, puis à transporter ces supports vers la machine destinataire. Cette méthode de transmission offre un large débit, à coûts restreints, mais le délai de transmission engendré est considéré comme déplorable.

Un autre moyen pour transférer des données consiste à relier les ordinateurs à l'aide de câbles jouant le rôle de support physique de transmission. Différents types de câbles sont disponibles, dont chacun d'eux a ses propres caractéristiques. Ces différentes caractéristiques sont exposées à l'annexe 3 (VOIR p. 82).

Les types de réseaux

On dissocie souvent les réseaux selon leur technique de transmission, soit les réseaux à diffusion où toutes les machines se partagent un seul canal de communication, soit les réseaux point à point qui sont formés d'un ensemble de connexions entre les machines prises deux à deux (liaisons point à point).

On peut également classer les réseaux selon leur taille.

Les réseaux locaux (LAN) sont des réseaux privés dont la taille ne dépasse pas quelques kilomètres. Ce sont souvent des réseaux à diffusion dont la topologie s'inscrit sous forme de bus ou d'anneaux.

Les réseaux longue distance (WAN) couvrent une zone géographique beaucoup plus importante, dont la taille est de l'ordre du millier de kilomètres. Leur structure est d'ordinaire composée d'un ensemble de LAN reliés entre eux par un sous-réseau. Ce sous-réseau est un réseau point à point, constitué d'un ensemble de routeurs connectés entre eux.

Les réseaux LAN

En pratique, les hôpitaux disposent souvent d'un ou plusieurs réseaux locaux (LAN) privés. Des normes d'implémentation ont été produites pour les réseaux LAN. On peut en analyser les plus importantes à l'annexe 4 (VOIR p. 83).

Les réseaux WAN

Les réseaux locaux permettent aux spécialistes de l'institution de communiquer entre eux, de transmettre des données au sein de l'établissement. Cependant, les besoins en communication dépassent généralement les frontières du réseau intra-hospitalier. C'est pourquoi il est souhaitable de pouvoir connecter les réseaux locaux vers un réseau extérieur, comme Internet qui reste le réseau des réseaux. Deux solutions sont possibles pour établir une telle connexion. Soit on utilise un ou deux routeurs qui possèdent des liaisons point à point louées vers des routeurs distants du réseau extérieur. Soit on souhaite accéder à un réseau extérieur en utilisant des liaisons point à point publiques auxquelles on parvient à l'aide d'un fournisseur d'accès.

Quelque soit la solution choisie, on doit emprunter des réseaux publics et donc utiliser les moyens de transmission offerts par leurs exploitants. Les réseaux publics englobent les réseaux téléphoniques, les réseaux ISDN, les réseaux de télédistribution, les réseaux ATM, les réseaux radiocellulaires et les réseaux satellites qui sont présentés à l'annexe 5 (VOIR p. 85).

Les protocoles

Pour que les machines, connectées par un média de transmission, puissent s'échanger de l'information, il faut leur permettre de communiquer via un protocole définissant des règles de communication. La plupart des communications dans les réseaux sont organisées en couches, chacune offrant certains services aux couches plus hautes. Evidemment, on observe diverses découpes en couches suivant les types de réseaux. Cependant, les hiérarchies de couches sont le plus souvent fondées soit sur le modèle OSI, soit sur le modèle TCP/IP. Aujourd'hui, on dispose d'une série de protocoles, chacun adapté à une couche dans un type de réseaux particulier et ayant chacun des performances qui leur sont propres. On peut sommairement identifier les protocoles suivants :

Les protocoles de la couche de liaison de données ont pour but de permettre la communication entre deux machines adjacentes, dans des réseaux point à point. Les protocoles les plus utilisés sont les suivants : SDLC, ADCCP, HDLC, PPP et SLIP. Ces deux derniers protocoles sont ceux utilisés dans la couche liaison de données de l'Internet.

Les protocoles de la couche de contrôle d'accès au canal ont pour objectif de permettre la communication entre deux machines connectées dans un même réseau à diffusion. On peut en citer les principaux : ALOHA, CSMA(/CD), DQDB et WDMA.

Les protocoles de la couche réseau sont chargés de transporter les données tout au long d'un chemin composé de diverses liaisons point à point, d'une machine source vers une machine destinataire. Le protocole le plus connu est, sans doute, le protocole IP utilisé dans le réseau Internet.

Les protocoles de la couche de transport fournissent plusieurs services dont le plus important est de permettre le transfert d'informations d'une machine émettrice à une machine réceptrice de manière fiable et économique, indépendante de la nature du ou des réseaux mis en place. On peut citer deux protocoles utilisés dans l'Internet : TCP et UDP.

Les protocoles de la couche application, quant à eux, permettent une grande variété de services tels que : l'échange de courrier (SMTP, POP3, IMAP), l'échange de fichiers (FTP, HTTP), etc.

Les moyens de sécurisation

Une autre considération est le problème de la sécurité. Un système de télémicroscopie fournit un ensemble de données qui sont archivées sur un support. La sécurisation de ces données, comprend le maintien de leur confidentialité, leur authenticité et leur intégrité. *La confidentialité rend compte du fait que seuls les utilisateurs habilités doivent pouvoir prendre connaissance des données* [Tanenbaum99]. Avec l'authenticité, on cherche à avoir la certitude que les données dont on dispose sont bien celles que l'on croit. L'intégrité signifie que les données dont on dispose n'ont pas été altérées.

Pour prétendre à la sécurité des données, il faut tout d'abord protéger physiquement les supports sur lesquels sont placées les données. En effet, il est stupide de penser à une sécurisation plus élaborée si les données sont archivées sur un support accessible à tous, comme par exemple un CD-ROM déposé sur une table ou le disque dur d'un ordinateur dont l'accès n'est pas restreint.

Une fois cette pré-condition établie, on dispose de différents moyens afin d'assurer la confidentialité, l'authenticité et l'intégrité des données, tels que l'identification des utilisateurs, l'utilisation de firewalls et la cryptographie mais également l'audit et les logiciels anti-virus.

2.5.6. La visualisation, l'édition et l'interprétation des documents

Les documents multimédia doivent pouvoir être également lus, édités et interprétés.

Pour construire un système d'édition de documents multimédia adapté à l'utilisateur, celui-ci doit fournir un ensemble d'opérations permettant la création, la modification, le stockage et la présentation des documents. On peut diviser ces opérations selon deux types :

- *Des opérations d'édition. Elles réalisent les opérations de création, de construction et de modification du document par un auteur. L'opération de composition d'un document consiste à y inclure des éléments multimédia de base, ensuite à spécifier des relations entre ces différents éléments. Ces relations peuvent être liées à leur organisation logique, leur disposition spatiale, ou encore à leur synchronisation temporelle.*
- *Des opérations de présentation. Elles permettent de restituer au lecteur le contenu du document une fois que son édition est achevée. Cette phase consiste à fournir au lecteur un ensemble de commandes permettant d'explorer ou de naviguer dans le document pour découvrir l'information qu'il contient à travers l'espace et le temps [Layaida97].*

Le paradigme WYSIWYG (What You See Is What You Get) est le principe de base sur lequel devrait être fondée la construction d'un système d'édition. Cette approche consiste à effectuer le processus d'édition selon un cycle d'action puis de perception, où à chaque étape, l'auteur réalise une action sur le document puis le système applique cette action et présente à l'auteur le document tel qu'il sera présenté au lecteur [JRT98]. On cherche donc à intégrer autant que possible les fonctions d'édition et de présentation dans un environnement unique pour que l'auteur puisse à tout moment vérifier le résultat de ce qu'il est en train de réaliser.

Cependant, à cause de la dimension temporelle, on ne peut mettre directement en œuvre le paradigme WYSIWYG dans une application d'édition de documents multimédia. En effet, cette dimension pose la question de la présentation d'un document lorsque l'auteur introduit des enchaînements temporels entre les médias. Il n'est pas pensable de présenter à l'auteur l'ensemble d'une séquence à chaque fois que celui-ci édite le document.

Donc, un outil d'édition de documents multimédia ne peut que se rapprocher du paradigme WYSIWYG en offrant, par exemple, une présentation lors de l'édition des dimensions logique, spatiale et hypermédia, puis une seconde présentation lors de l'édition de la dimension temporelle [JRT98].

Un autre exemple d'architecture possible pour un système d'édition de documents multimédia est de construire différentes vues d'un même document, représentant l'ensemble des dimensions de ce document. On peut identifier quatre vues : la vue de présentation, la vue du scénario, la vue des objets et la vue du storyboard.

La vue du storyboard est une suite de prises de vue du document, miniaturisées et capturées à des points stratégiques du temps, entre lesquelles l'intervalle peut être paramétré. Cette vue permet de naviguer dans la dimension temporelle et de sélectionner un des points stratégiques du document. Elle permet aussi d'éditer la dimension temporelle, par exemple, en ajoutant une prise de vue ou en faisant varier l'ordre ou les intervalles de temps entre ces prises de vue. La vue de présentation permet de présenter en détails le document au point du temps sélectionné. Elle permet également d'éditer la dimension spatiale, par exemple, en ajoutant

différents objets ou en les déplaçant. Bien entendu, ces opérations doivent être le plus proche possible de l'approche WYSIWYG.

La vue des objets offre un aperçu des objets contenus dans le document et permet d'obtenir une description de ces mêmes objets. Elle a pour but d'offrir une perception globale du contenu du document.

Enfin, la vue du scénario a pour but de donner à l'auteur une perception globale de l'organisation temporelle du document. Elle répartit l'apparition des objets sur l'axe du temps, dont les segments correspondent aux intervalles entre les prises de vue du storyboard.

L'intérêt de cette architecture réside également dans le fait que ces quatre vues sont interdépendantes et qu'une modification dans une de ces vues peut entraîner des modifications dans les autres vues [Layaida97].

Cependant, il ne faut pas négliger l'intérêt d'autres architectures car, en effet, cette architecture n'est qu'un exemple parmi tant d'autres.

De façon générale, un système d'édition de documents multimédia doit répondre à certaines exigences.

La création d'un document se fait généralement par un enrichissement progressif de son contenu et de sa structure. Chaque opération d'édition effectuée par l'auteur doit lui permettre de voir immédiatement son effet à la présentation.

Il faut diminuer au maximum, la distance entre le travail d'interprétation et la composition multimédia [Emim02]. Lorsqu'un utilisateur édite un document, il doit immédiatement comprendre quelles sont les conséquences des opérations qu'il effectue sur ce document.

Pour chaque opération d'édition effectuée par l'auteur, le système d'édition doit vérifier sa cohérence par rapport à l'état courant du document.

Il est nécessaire de prendre en compte les quatre aspects d'un document multimédia : son organisation logique, sa synchronisation temporelle, sa présentation spatiale et son interconnectivité intra- et inter- documents [Layaida97].

L'auteur doit également pouvoir avoir une vue globale du document, qui peut l'aider à mieux gérer la complexité d'un document multimédia. Pour les documents traditionnels, cette exigence peut être satisfaite par une table des matières, mais dans le cas de documents multimédia, cela nécessite peut-être d'autres moyens [Layaida97].

Ces différentes exigences attendues d'un système d'édition multimédia proviennent d'une part du besoin de construire des applications interactives simples d'utilisation et d'autre part de permettre une représentation du document riche du point de vue sémantique et échangeable entre différentes applications multimédia [Layaida97].

Deuxième partie : Matériel et méthode

3. La télémicroscopie à l'UCL Mont-Godinne

3.1. L'infrastructure disponible

3.1.1. La station d'acquisition

Pour développer le système de télémicroscopie, on dispose d'un microscope « AX 70 d'Olympus » dont les composants principaux sont motorisés, principalement la platine et le pivot des objectifs. Trois objectifs de grossissements différents sont installés sur le pivot : un objectif 20X, un objectif 40X et un objectif 100X. Ce microscope supporte évidemment le contrôle à distance pour les principales fonctions des composants motorisés, grâce à l'unité de contrôle « U-MCB d'Olympus » avec laquelle l'ordinateur peut communiquer via une connexion par port série, respectant le standard « rs232 ». Par exemple, l'ordinateur peut envoyer un message à l'unité de contrôle via la connexion. Si ce message est correctement formaté et indique l'intention de déplacer la platine alors l'unité de contrôle répercute cet ordre en déplaçant effectivement la platine grâce aux moteurs. On dispose également d'un autofocus hardware « U-AF d'Olympus » associé à une caméra analogique « Sony DXC-950 » placée sur le microscope afin de capturer les images. Cette caméra est reliée à une carte de digitalisation « Matrox Meteor RGB » qui est installée sur l'ordinateur utilisé pour l'acquisition. La machine d'acquisition dispose d'un disque dur de 60 Go, de 640 Mo de mémoire RAM, d'un processeur de 730 Mhz. Le système d'exploitation installé sur cette machine est le système « Windows NT 4.0 ». La machine est connectée en permanence au réseau local et possède un accès à l'Internet.

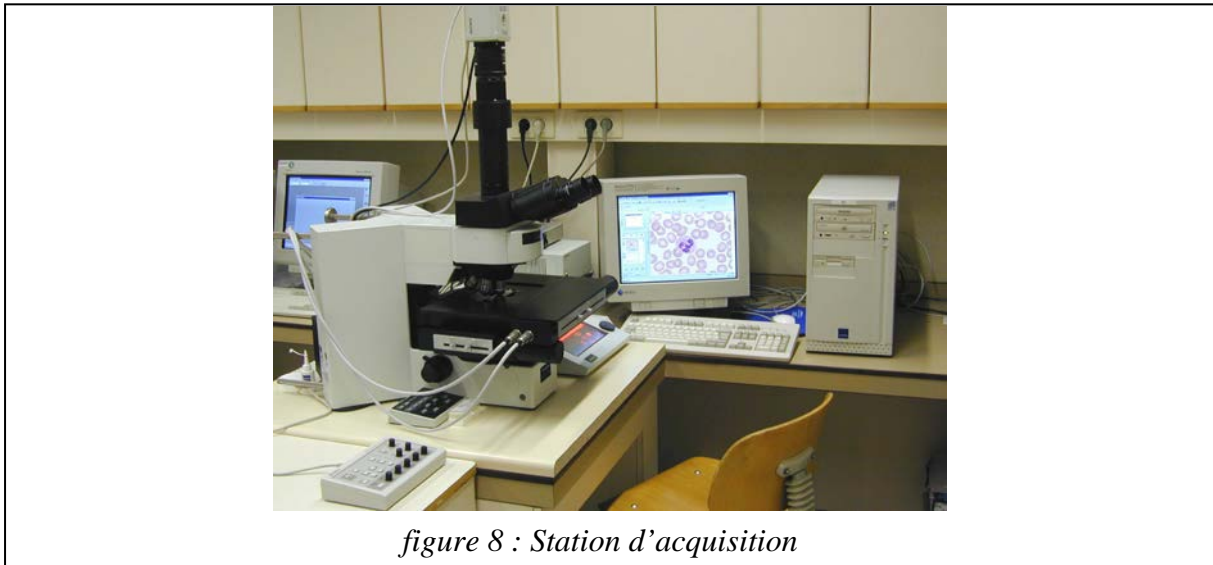


figure 8 : Station d'acquisition

3.1.2. Le logiciel « Analysis »

La machine d'acquisition exploite le logiciel « Analysis Pro 3.0 » (Soft-Imaging Software GmbH). Ce logiciel est conçu pour l'acquisition, l'analyse, le traitement et l'archivage d'images. De plus, il possède un module, nommé « AX70 » qui fournit l'interface nécessaire au contrôle à distance des fonctions disponibles du microscope « AX 70 d'Olympus ». Ce module procure une librairie de fonctions C correspondant aux fonctions disponibles du microscope mais également la possibilité de saisir et fixer les paramètres accessibles du microscope.

Le logiciel est également pourvu d'un composant, appelé « Imaging-C », qui permet à l'utilisateur, grâce à un interpréteur incorporé, de personnaliser l'application par la création de modules personnels (sous forme de code C). Cet interpréteur permet l'utilisation de variables, de structures de contrôles (for, while, if, ...), l'importation de DLL, l'emploi de tous les objets et les fonctions de traitement d'images disponibles dans « Analysis » et également l'importation de fonctions entre les modules développés tels que le module « AX70 » ou les modules créés par l'utilisateur. « Imaging-C » est basé sur le ANSI C mais fournit également une extension de ce standard, composée de nouvelles méthodes pré-définies. Il est également compatible avec le « MS Windows Software Development Kit » (SDK) permettant la programmation de nouvelles interfaces graphiques dans l'environnement « MS Windows ». L'utilisation d'une fonction développée dans un module C peut être intégrée facilement dans l'interface d'« Analysis » (VOIR annexe 6, p. 87) en associant cette fonction à un bouton ajouté dans une barre de boutons ou à un item dans un menu, grâce à une simple ligne de code placée dans ce module.

3.1.3. Le logiciel « Télémis »

La description

L'UCL Mont-Godinne possède déjà un système de télémédecine interne permettant de consulter et d'échanger les images en provenance des plateaux techniques d'acquisition tels que la « Magnetic Resonance » (MR), la « Computed Tomography » (CT), la « Positron Emission Tomography » (PET), etc. Ce système est baptisé « Télémis » et est développé par une spin-off de l'université catholique de Louvain, ayant le même nom. On peut décrire le système « Télémis » comme un ensemble logiciel composé de trois modules spécifiques (*figure 9*).

Le premier module, appelé « Télémis-Acquisition », réalise l'acquisition des images depuis les modalités. Les images doivent simplement être envoyées sous le format DICOM et via le protocole DICOM, depuis les plateaux techniques vers le serveur « Télémis-Acquisition ». Elles sont ensuite converties en un document multimédia comprimé et crypté. Enfin, le module détermine, en fonction de l'application de filtres prédéfinis sur les informations contenues dans les images, la boîte aux lettres personnalisée d'images, appelée « Télémis-Pim », dans laquelle le document est envoyé.

Le second module, appelé « Télémis-Serveur », stocke le document expédié par le module d'acquisition, dans la boîte aux lettres électronique du destinataire souhaité. Chaque destinataire possède donc une boîte, en réalité un répertoire, dans lequel sont placées toutes les images lui étant destinées. Ce module permet également à un utilisateur d'accéder à sa boîte aux lettres électronique. Si le client est authentifié, le module lui accorde l'accès à sa « Télémis-Pim » et permet ainsi aux images stockées dans celle-ci d'être envoyées vers l'ordinateur client, ordinateur sur lequel est installé le dernier module.

Le dernier module, appelé « Télémis-Reception », permet à des utilisateurs authentifiés par un mot de passe et une clé d'identification personnelle, de télécharger et visualiser la liste des documents sécurisés qui leur sont adressés par le biais d'une première interface (VOIR annexe 7, p. 88). Cette interface permet, par un simple clic, la sélection et la consultation d'un des documents de la liste. Pour ce faire, le module charge à l'écran les images contenues dans le document sélectionné, tout en les décompressant et décryptant simultanément. Ces images sont affichées dans une deuxième interface (VOIR annexe 7, p. 88), sous forme de galerie. Des outils, accessibles via cette deuxième interface, sont intégrés à ce module afin de permettre la manipulation et le traitement des images reçues. On peut citer comme exemple, la possibilité d'utiliser une loupe ou un zoom, de changer le contraste et la luminosité, de

signaler des régions d'intérêts. On dispose même de la possibilité d'effectuer des visualisations coopératives à distance c'est-à-dire de partager l'écran entre deux postes distants et permettre d'interagir à partir de ces deux postes.

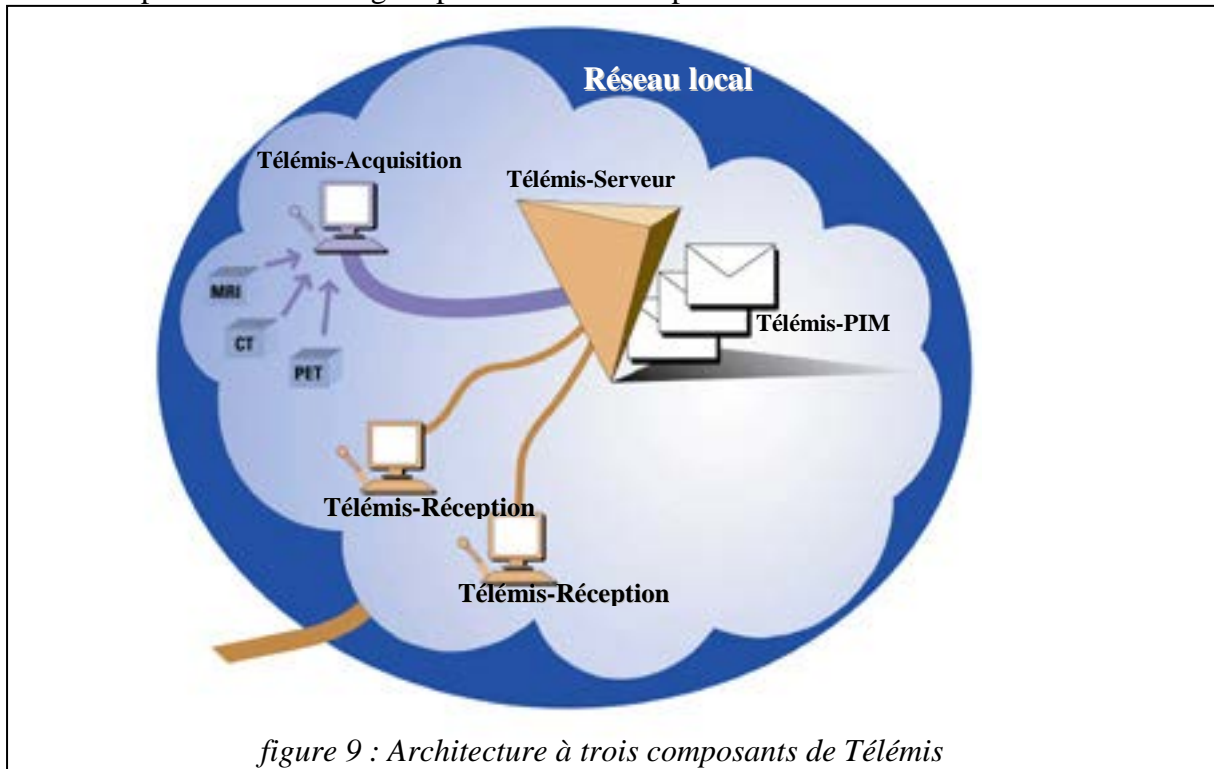


figure 9 : Architecture à trois composants de Télémis

Les caractéristiques propres à ce système diffèrent relativement peu de celles nécessaires à un système statique de télémicroscopie. On peut donc s'inspirer, voir même réutiliser les éléments de ce système pour donner une structure et une architecture de base à la communication entre médecins dans le domaine de la microscopie.

Les avantages et inconvénients

Le système « Télémis » offre quelques avantages non négligeables tels que la compression automatique, la sécurisation via l'authentification et le cryptage des documents, des moyens complets et performants de visualisation et de manipulation d'images.

Cependant il ne faut pas perdre de vue que ce système comporte également des lacunes.

Premièrement, les documents multimédia utilisés dans ce système, ne représentent en fait qu'un regroupement d'images dans une structure ne précisant aucune dimension logique, spatiale, temporelle ou hypermédia entre les éléments. Outre cela, ces documents respectent un format propriétaire, ce qui rend leur lecture impossible par d'autres applications. Ces constatations mettent en évidence la pauvreté du type de documents employés par le système « Télémis ».

Deuxièmement, les fonctionnalités propres à la manipulation et au traitement des images ne sont pas identiques à toutes les disciplines. Or les outils développés dans le système « Télémis » sont dédiés à la radiologie. Il faut donc faire évoluer ce produit afin de l'adapter aux exigences de la microscopie.

Enfin, le système « Télémis » n'offre aucune option d'édition des documents. Un utilisateur peut télécharger à volonté les documents qui lui sont adressés mais en aucun cas il ne pourra changer le contenu du document se trouvant sur le serveur.

3.1.4. L'ADICAP

La description

L'ADICAP est l'Association pour le Développement de l'Informatique en Cytologie et Anatomie Pathologique. Cette association regroupe des experts en cytologie et en anatomopathologie. Elle est impliquée dans la mise au point d'un système d'échanges de documents entre spécialistes en cytologie et en anatomie pathologique, qui porte son nom et auquel l'UCL Mont-Godinne s'est abonné. Ce système a été développé par le CRIHAN, Centre de Ressources Informatiques de Haute-Normandie. Le mécanisme de ce système de télétransmission de documents est simple.

Premièrement, il faut générer un fichier zip contenant toutes les informations du document et respectant les spécifications de l'ADICAP (figure 10). Un fichier zip regroupe, en les compressant, un ensemble de fichiers ayant chacun sa place dans une arborescence. L'arborescence est totalement libre dans sa structure et dans son appellation. Néanmoins, le nom du répertoire racine devra être identique au nom du fichier zip, qui ne doit contenir aucun espace. Ce répertoire devra contenir trois fichiers du même nom dont les extensions seront respectivement « eml », « htm » et « ini ».

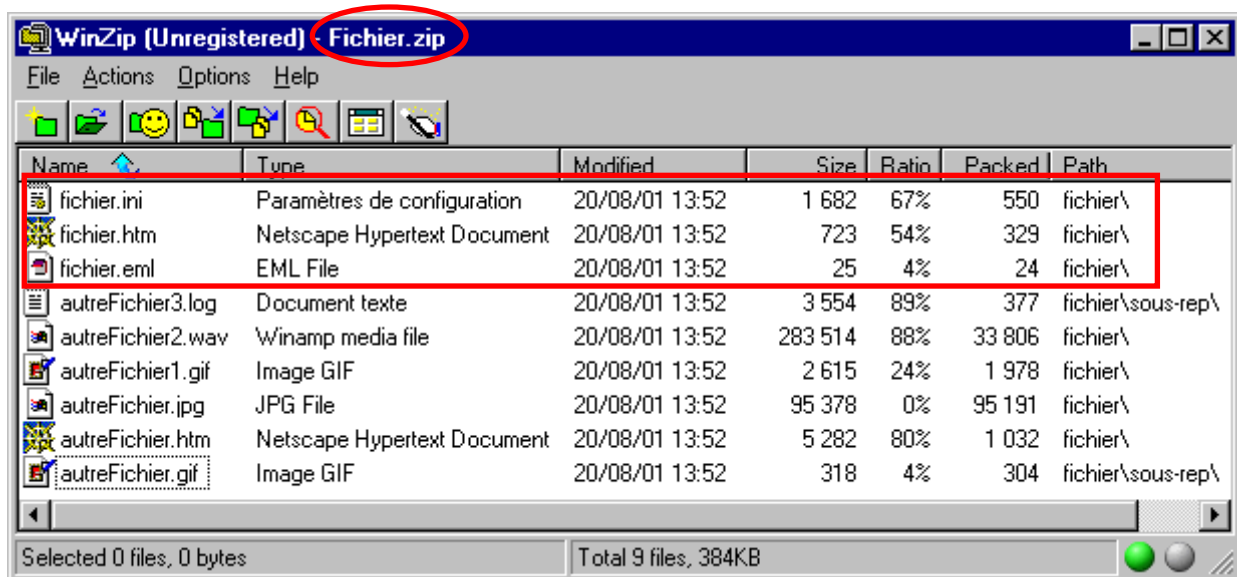


figure 10 : Exemple de fichier zip respectant la spécification de l'ADICAP (présenté par le logiciel WinZip)

Le fichier d'extension « eml » est un fichier texte contenant les adresses électroniques (e-mail), valides, des personnes qui auront accès au document. Chaque ligne contient une seule adresse.

Le fichier d'extension « htm » est en fait la page d'accueil du document qui est présenté au destinataire. Ce fichier doit respecter la norme HTML version 3. Il est également recommandé de veiller à n'utiliser dans les pages HTML que des liens relatifs afin qu'ils restent valides une fois ces fichiers installés sur le serveur de l'ADICAP.

Le fichier d'extension « ini » est un fichier texte contenant un descriptif du document envoyé. Ce fichier est divisé en deux parties : une partie commune à tout le document et une partie spécifique à chaque image contenue dans le document (figure 11).

[Commun]
Date = date
Heure = heure
Titre = 80 alpha
Comment = illimité, codé html
Site_Emetteur = nombre entier **Chaque utilisateur de l'ADICAP appartient à un site identifié par un nombre entier**
Liste =Site_NumérisationX/ImageX;Site_NumérisationX/ImageY ;...

[Site_NumérisationX/ImageX]
Createur = 25 alpha **Propriétaire de l'image**
Date_Modif = date
Date_User = date
Diapo_Lame = D ou L **Selon codification ADICAP**
Reference = 20 alpha **Laboratoire producteur de l'image**
Responsable = 25 alpha **Individu ou groupe producteur de l'image**
Document_Type= 01 à 16 **Selon codification ADICAP**
Prelevement = 1 alpha **Selon codification ADICAP**
Technique = 1 alpha **Selon codification ADICAP**
Organe = 2 alpha **Selon codification ADICAP**
Lesion1 = 8 alpha **Selon codification ADICAP, chaîne "NULL" si vide**
Lesion2 = 8 alpha **Selon codification ADICAP, chaîne "NULL" si vide**
Lesion3 = 8 alpha **Selon codification ADICAP, chaîne "NULL" si vide**
Legende = 255 alpha
Comment = 255 alpha
Source_Image = 255 alpha **Chemin relatif de l'image**
Source_Icone = 255 alpha **Chemin relatif de l'imagette**

figure 11 : Organisation du fichier d'extension « ini » contenant le descriptif du document

Après avoir engendré le fichier zip, il suffit d'envoyer ce fichier sur l'ordinateur du serveur de l'ADICAP via le protocole FTP. Cette opération nécessite l'authentification de l'utilisateur par son nom et son mot de passe que le secrétariat de l'ADICAP lui a communiqué lors de son enregistrement au cours duquel un compte a été créé à son attention, sur le serveur de l'ADICAP (www.adicap.asso.fr ou 195.221.21.131). Après s'être authentifié, l'utilisateur est positionné sur son répertoire personnel où il dépose le fichier zip. La gestion des comptes (figure 12) est effectuée de telle manière que le répertoire personnel d'un utilisateur se trouve dans un répertoire identifié par le numéro du site émetteur, qui réside lui-même dans le répertoire racine du serveur.

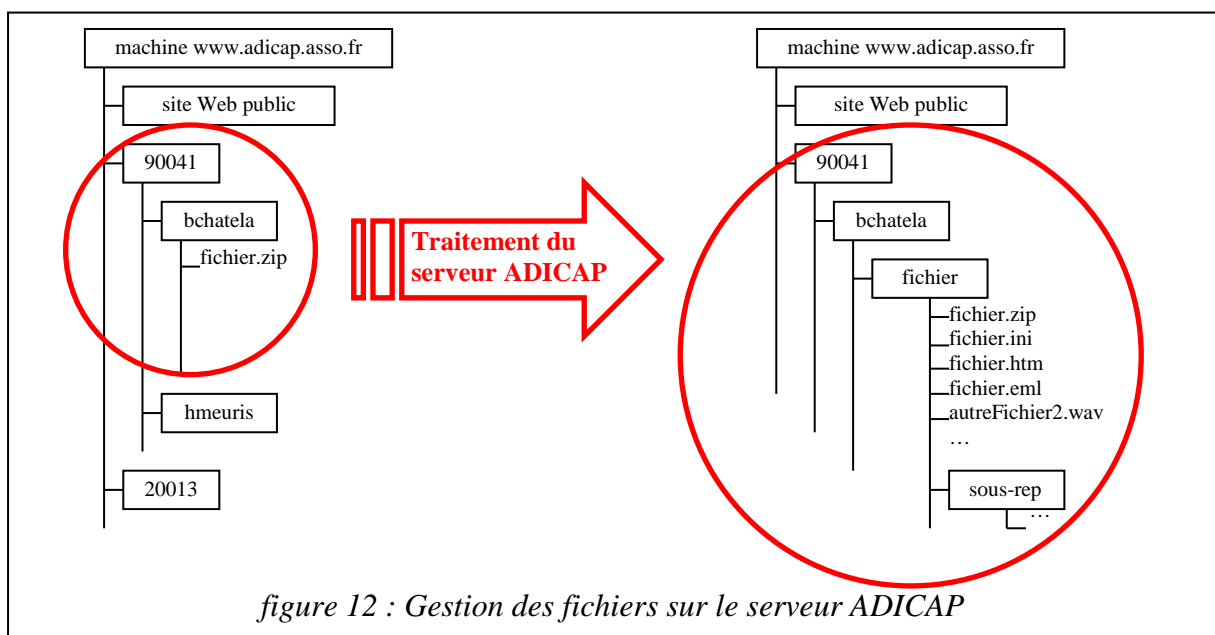


figure 12 : Gestion des fichiers sur le serveur ADICAP

Un fois le fichier zip déposé sur la machine du serveur, il est pris en charge par ce dernier (figure 13).

Premièrement le serveur vérifie la cohérence du fichier. L'incohérence du fichier, due à sa composition, à une mauvaise réception ou à la présence d'espaces dans le nom, entraîne sa destruction et l'envoi d'un message à l'expéditeur.

Ensuite, si le fichier est cohérent, le serveur de l'ADICAP le décompresse dans un répertoire dont le nom est le même que celui du fichier exempt de son extension « .zip ». Les éléments du nouveau document sont alors mis à disposition sur le serveur HTTP de l'ADICAP en affectant les droits de lecture adéquats aux destinataires.

Pour finir, le serveur de l'ADICAP envoie des courriers électroniques aux destinataires pour les avertir de la mise à disposition du nouveau document. Ce message électronique a pour sujet le site émetteur suivi du nom du document et indique dans son corps l'URL à laquelle ce document est disponible.

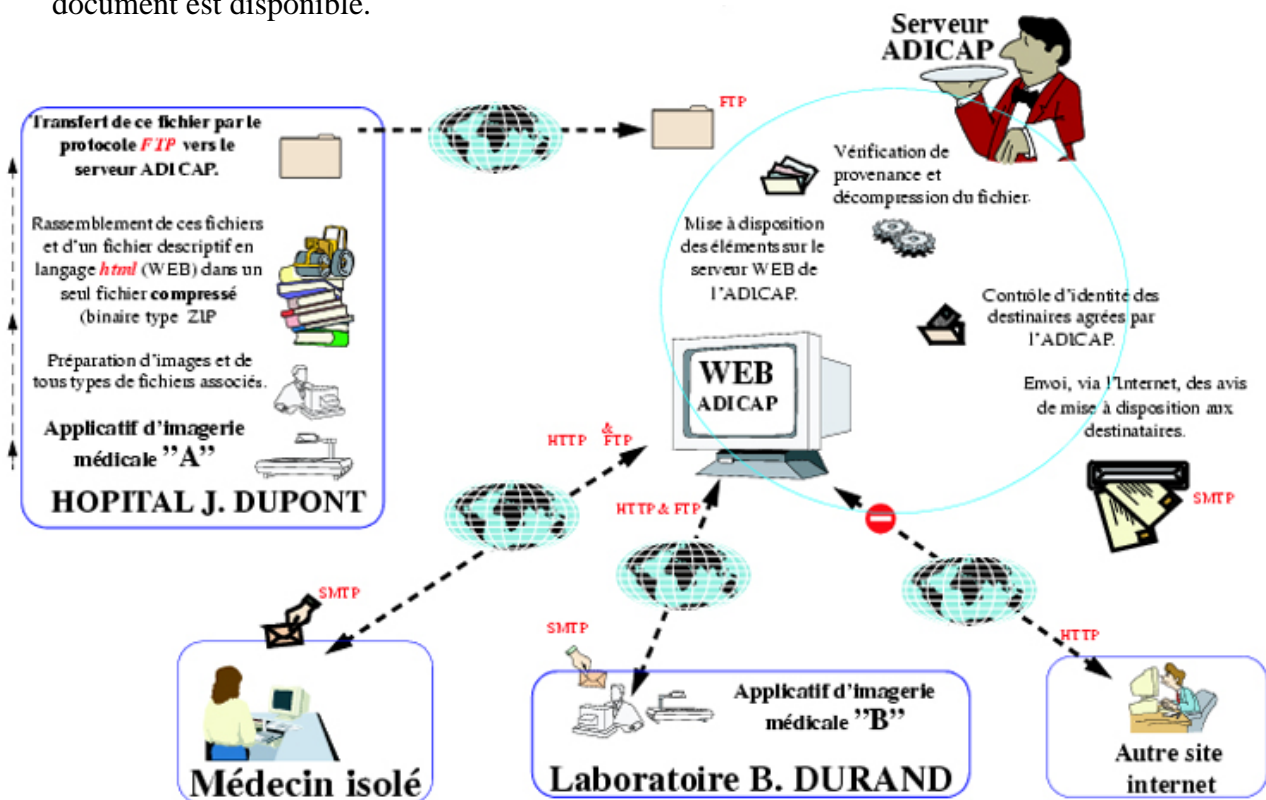


figure 13 : Mécanismes d'échange de documents via le serveur ADICAP

Les avantages et inconvénients

L'ADICAP fournit un moyen simple et efficace de demande d'expertise en temps différé. D'abord, il offre la possibilité d'une consultation simultanée d'un document par plusieurs spécialistes. De plus, il s'appuie sur des protocoles de communication (FTP, HTTP et SMTP) et un format de documents (HTML) standards, ce qui rend possible une grande accessibilité et une certaine interopérabilité dans un domaine où l'on doit faire face à un ensemble hétérogène d'applications.

Cependant, ce système a également des défauts. L'utilisation du format HTML et du protocole HTTP contraint les utilisateurs à employer un web-browser, peu adapté à la cytologie, pour visualiser le document. En outre, dans ce système, les données transitent par l'Internet et aucun moyen de sécurisation n'est mis en œuvre lors des échanges entre le site émetteur et l'ADICAP via le protocole FTP et ni lors des échanges entre l'ADICAP et le site destinataire via le protocole HTTP. L'ADICAP n'effectue aucun contrôle en vue de détecter

les fichiers porteurs de virus et rejette cette responsabilité sur l'utilisateur. Enfin, les échanges de documents effectués à l'aide du système de l'ADICAP sont limités uniquement aux spécialistes inscrits. Cette politique réduit donc considérablement, l'accessibilité engendrée par les choix technologiques [ADICAP00].

3.2. La cytologie hématologique à l'UCL Mont-Godinne

3.2.1. La situation initiale

L'analyse cyto-hématologique à l'UCL Mont-Godinne obéit à la description qui a été faite dans la première partie de ce mémoire (1.2. Description d'une analyse). Cette description est cependant accompagnée de quelques particularités. Il est donc propice de détailler l'analyse telle qu'elle est effectuée dans le laboratoire de l'UCL Mont-Godinne. La décomposition du traitement global de l'analyse fait apparaître quatre phases séparées chacune par l'intervention d'un individu et fournit le modèle de structuration des traitements suivants (*figure 14*) :

1) Nom : Enregistrement du patient

Définition : Au moment de la réception des échantillons, le secrétariat du laboratoire associe le sang et le patient en ouvrant un dossier dans le système central de gestion des dossiers de patients « GesPower ». Au même moment ce dernier crée alors une demande d'analyse qui est associée au dossier par un numéro unique composé du secteur d'activité du laboratoire, de l'année, du numéro de la semaine et du numéro du jour. A partir de ce numéro unique, un code barre est édité et sera placé sur l'échantillon de sang et sur tout autre dérivé de cet échantillon.

Décomposition en fonctions : - Enregistrement du patient

2) Nom : Analyse cytométrique

Définition : Chaque demande d'analyse, générée dans la première phase, parvient à un poste de travail. La deuxième phase peut alors commencer. Un employé du laboratoire satisfait, une à une, les demandes qui s'accumulent sur le poste de travail. A l'aide du cytomètre, il analyse le sang de l'échantillon correspondant à la demande qu'il est en train de traiter, afin de déterminer si le sang du patient est sain ou non et donc de donner une réponse négative ou positive. Le cytomètre renvoie automatiquement le résultat au système central de gestion des dossiers.

Si l'analyse du sang aboutit à une réponse négative, le système ajoute le résultat au dossier du patient et le clôture. Dans le cas contraire, le système indique qu'une observation microscopique est nécessaire. Il faut aussi remarquer que seuls les cas de réponse positive peuvent être erronés. Seuls ces cas nécessitent une analyse microscopique approfondie afin d'écarter les cas de réponse faussement positive.

Décomposition en fonctions :
- Analyse cytométrique
- Clôture du dossier
- Indication d'observation

3) Nom : Prise de connaissance des demandes d'analyse microscopique

Définition : Cette phase consiste en une prise de connaissance des demandes d'analyse qui ont conduit à un résultat positif lors de l'analyse cytométrique. Cette phase est réalisée par une simple requête sur le système central de gestion des dossiers. Ce dernier fournit un fichier codé en caractère ASCII et énumérant les demandes sous forme d'une suite de lignes, chacune indiquant le numéro unique, le nom du patient, le prénom du patient, sa date de naissance et la date de prélèvement.

Décomposition en fonctions : - Prise de connaissance des demandes

4) Nom : Analyse microscopique

Définition : Pour chaque ligne du fichier obtenu lors de la phase précédente, un laborantin étale le sang de l'échantillon correspondant au numéro, sur une lame porte objet. Il place également sur cette lame un code barre afin d'éviter toute méprise par rapport à l'association de la lame au dossier du patient. La lame est ensuite placée sur la platine d'un microscope et un observateur expérimenté examine manuellement l'étalement afin d'émettre un diagnostic définitif. Une fois que l'observateur a déterminé son diagnostic, il ajoute ces résultats au dossier du patient correspondant et clôture ce dernier. Comme on l'a exprimé dans la première partie, bien que l'observateur soit expérimenté, il doit pouvoir communiquer avec d'autres experts pour se forger un avis pertinent.

Décomposition en fonctions :
- Analyse microscopique
- Echange d'avis
- Clôture du dossier

3.2.2. Les objectifs

L'objectif principal est de développer, à partir de la situation actuelle, un système de télémicroscopie permettant la production d'images numériques de qualité suffisante et la mise en place d'un réseau de consultations et d'échanges de ces images. Ce système doit évidemment permettre de jouir de l'ensemble des avantages qui ont été énumérés dans la première partie de ce mémoire.

Un objectif supplémentaire est de pouvoir, à partir de huit lames porte-objet, constituer de manière automatique, pour chacune de ces lames, une galerie d'images, qui se restreint à un rassemblement d'un nombre défini d'images représentatives de l'échantillon. Pour le sang périphérique, on estime le nombre d'images nécessaires entre 50 et 100. L'utilisation de galeries offre plusieurs avantages : la possibilité de comparer les cellules les unes aux autres, la comparaison aisée de deux galeries de cas cliniques différents [Mercier00].

Cet objectif implique l'automatisation, dans le système développé, de la sélection, de l'acquisition et du stockage des images numériques mais également de l'intégration des images et des données cliniques sous forme de documents et de la transmission de ceux-ci. Cependant un mode de sélection d'images automatiques peut mener à des interprétations erronées du cas analysé. C'est pourquoi, la structuration des traitements doit être adaptée en ajoutant la possibilité d'une analyse à posteriori lorsqu'il subsiste un doute lors de l'analyse automatisée. Cette analyse à posteriori est effectuée manuellement sur le microscope.

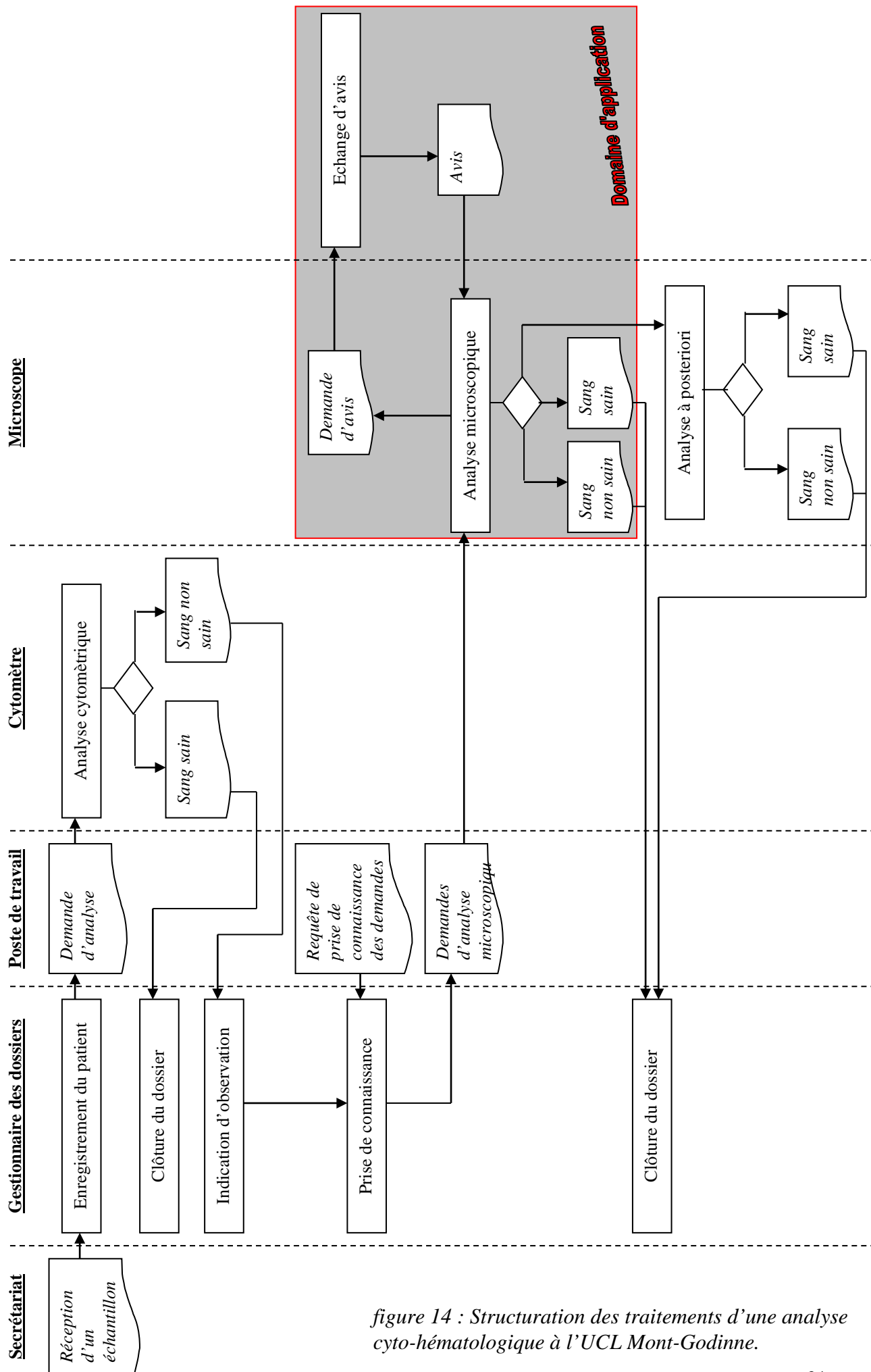


figure 14 : Structuration des traitements d'une analyse cyto-hématologique à l'UCL Mont-Godinne.

3.3. Le type du système à développer

Etant donné, la volonté de développer un système de télémicroscopie utilisant le système « Télémis » et l'objectif d'automatisation, il est sans doute plus favorable de développer un système recouvrant une forme statique.

De plus un système dynamique nécessite un large débit. Or seul le réseau local permet d'obtenir le débit adéquat pour ce type de systèmes. Mais le développement d'un tel système dans un réseau local n'apporte aucun avantage puisque l'observateur est localisé dans un environnement proche et peut donc facilement accéder au microscope lui-même. Cependant, il faut garder à l'esprit que l'utilisation d'un système dynamique reste un élément qui semble important à offrir.

On décide donc de développer un système statique (*figure 15*) dans lequel la station d'acquisition permet de réaliser la sélection et l'acquisition des images numériques nécessaires à l'analyse. Ces images sont stockées sur le disque local pour être à posteriori transférées sur CD car aucun système d'archivage n'est encore mis en place à l'UCL Mont-Godinne. Ces mêmes images doivent également être transmises au serveur du système « Télémis ».

Par la suite, elles doivent pouvoir être visualisées soit sur la machine d'acquisition en local ou sur toute autre machine via l'application cliente de « Télémis ».

Enfin, il ne faut pas négliger les besoins d'expertise et d'échange d'avis qui peuvent être améliorés par l'utilisation du système de télémicroscopie développé. Il faut donc permettre la transmission des résultats vers l'ADICAP ou via e-mail à tout autre expert possédant une adresse.

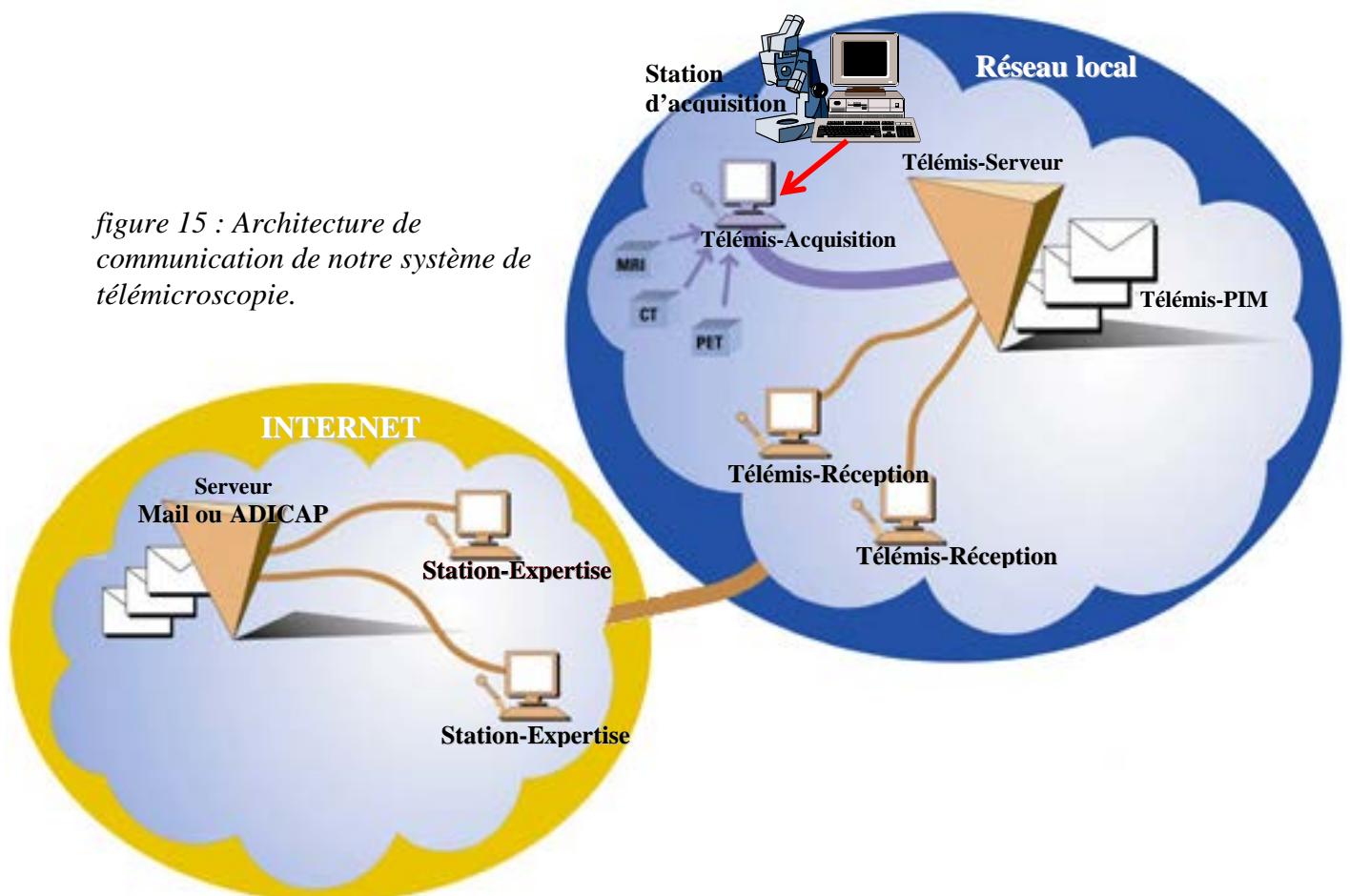


figure 15 : Architecture de communication de notre système de télémicroscopie.

3.4. La structuration fonctionnelle du système à développer

3.4.1. La sélection des images

Le système à développer devra proposer deux modes de sélection d'images.

Le premier mode de sélection consiste à laisser l'utilisateur manuellement déplacer la platine afin de naviguer dans la zone d'observation et de sélectionner les images qui l'intéressent.

Le deuxième mode de sélection est réalisé dans le cadre de la création automatisée d'une galerie d'images (*figure 16*). Ce mode de sélection consiste à balayer automatiquement, à l'objectif 20X, une zone de la lame porte-objet grâce à la platine motorisée. Pour chaque champ de la zone balayée, on va détecter automatiquement des régions d'intérêt à partir de paramètres, afin de répertorier leurs coordonnées sur l'axe x et y. La détection des régions d'intérêt, en l'occurrence des cellules nucléées spécifiques, se base sur deux paramètres : la couleur et la taille de leur noyau. Donc pour détecter un type de cellules nucléées spécifiques, on va simplement détecter l'ensemble des particules du champ observé, qui se trouvent dans un intervalle de teinte, de saturation et d'intensité de lumière particulier. Une fois ces particules détectées, il suffit de conserver uniquement celles qui forment des groupes de taille déterminée. Ces groupes de particules localisent les régions d'intérêt espérées.

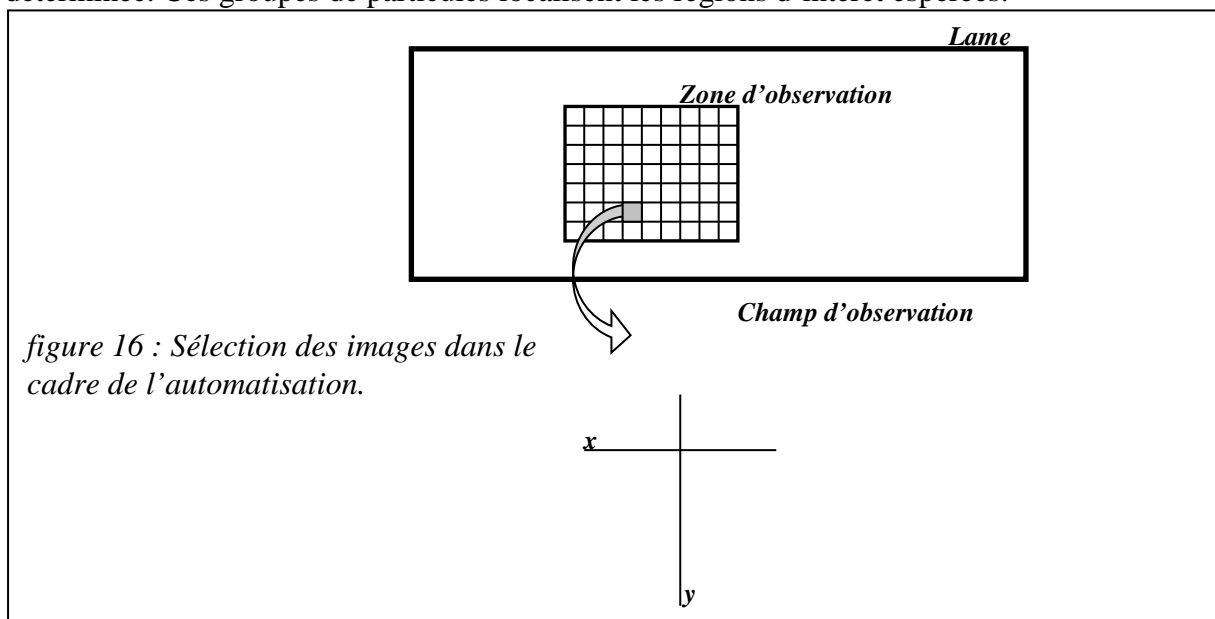


figure 16 : Sélection des images dans le cadre de l'automatisation.

3.4.2. L'acquisition des images

Le système à développer devra proposer deux types d'acquisition : l'acquisition manuelle qui est effectuée dans la logique du premier mode de sélection et l'acquisition automatique qui se combine au second mode de sélection.

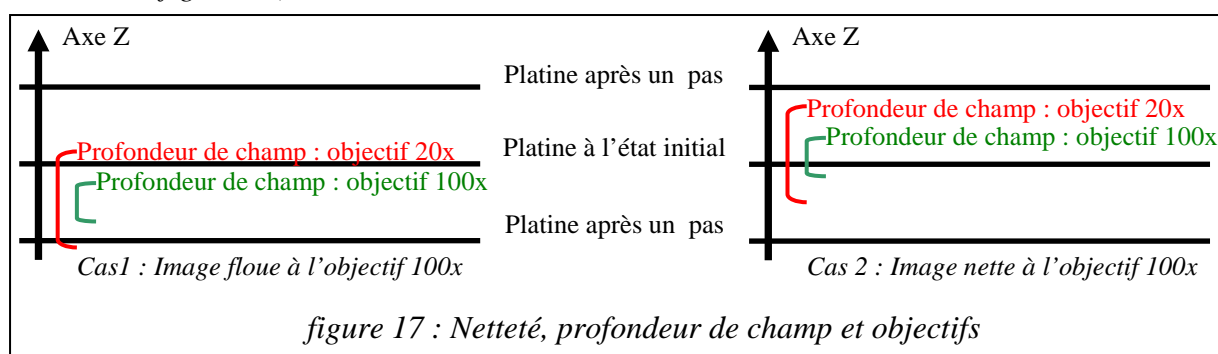
L'acquisition manuelle permet à l'utilisateur de conserver l'image sélectionnée dans un intervalle de mémoire temporaire.

L'acquisition automatique consiste à positionner la platine aux coordonnées répertoriées lors de l'étape de sélection et à capturer à l'objectif 100X, les images dont le centre correspond aux coordonnées.

Quel que soit le mode d'acquisition, celle-ci est effectuée à l'aide de la caméra « Sony DXC-950 » et la carte de numérisation. La caméra assure une haute qualité d'image, soit une définition de 755 sur 503 pixels, grâce à son capteur 3CCD qui est bien le facteur contraignant en terme de définition, puisque la définition de la carte de numérisation atteint 764 sur 573 pixels.

Enfin, il faut également remarquer qu'une mise au point précise de l'image avant sa capture est nécessaire afin d'obtenir sa netteté. La netteté de l'image est obtenue lorsque le contraste, différence du niveau de gris entre deux points qui composent l'image, est maximum. Autrement dit, on obtient la netteté lorsque deux points très proches dans l'image ont un niveau de gris le plus différent possible.

Pour effectuer la mise au point, on a tenté de réaliser un autofocus software en prenant pour chaque capture d'image une série d'images à différents niveaux de hauteur de la platine (axe z) et en conservant l'image dont le contraste est le plus élevé. Cependant, cette tentative a échoué car la profondeur de champ pour l'objectif 100X, qui est l'écart sur l'axe z pour lequel l'image est nette, est trop petite par rapport au pas de déplacement de la platine sur le même axe. Dès lors, il se peut qu'une image ne soit pas nette bien que l'autofocus software ait fonctionné (*figure 17*). On utilise donc un autofocus hardware.

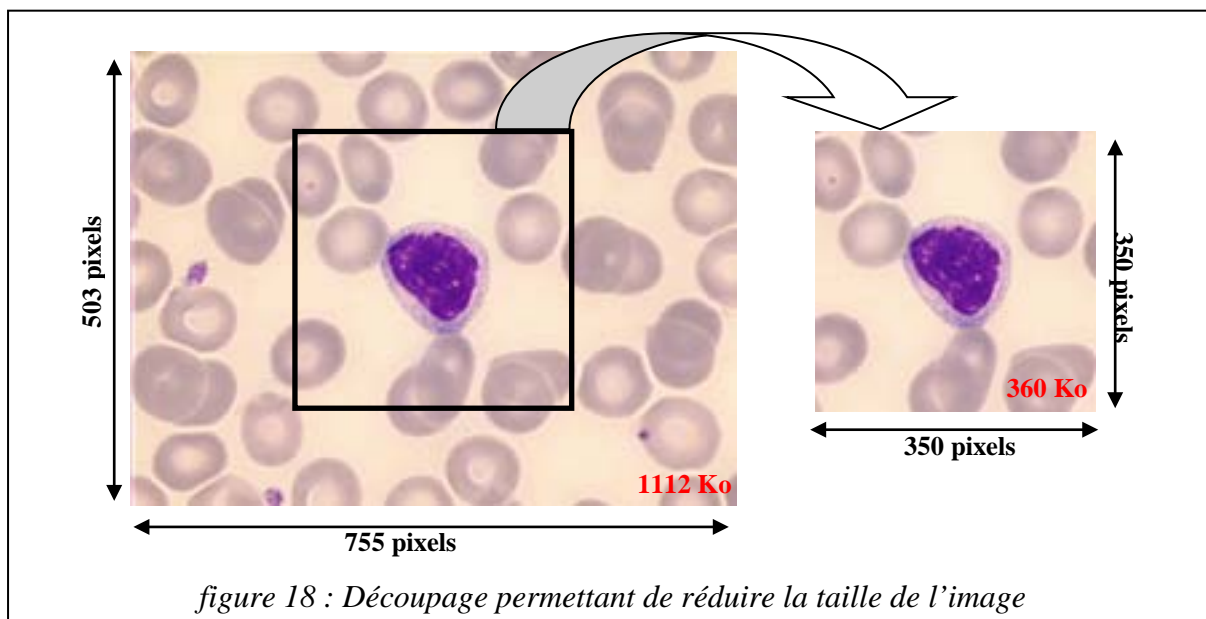


3.4.3. Le stockage des images

Si on considère l'objectif d'automatisation de la création de huit galeries comportant au minimum 50 images, on obtient un ensemble de 400 images à stocker. Or, les images acquises ont une définition de 755 sur 503 pixels et sont des images couleurs codées sur 24 bits. Donc, chaque image nécessite 1112 Ko. La quantité totale de mémoire nécessaire pour un cas clinique est de 54 Mo et pour l'ensemble des huit galeries, elle est estimée à 432 Mo, ce qui représente un espace non négligeable.

Pour le stockage, on a malgré tout choisi de conserver des images non compressées, à haute résolution et à haute qualité de pixel, car comme on l'a exprimé dans la première partie, toute tentative de réduction de la taille mémoire nécessaire entraîne une réduction de la qualité des images. Or, on peut toujours obtenir une image de mauvaise qualité à partir d'une image de bonne qualité, mais l'inverse n'est pas du tout concevable.

Cependant, on a remarqué que lors de la capture des images, une grande partie de celles-ci n'a aucun intérêt médical. A partir de cette constatation, l'idée d'effectuer un découpage (*figure 18*) est née, afin de réduire les images de 755 sur 503 pixels à des images de 350 sur 350 pixels de même résolution. Cette astuce permet de réduire la quantité de mémoire exigée pour le stockage d'une image à 360 Ko, soit 18 Mo par cas clinique et 144 Mo pour les huit galeries.



Pour stocker les images, on utilisera, dans un premier temps, le disque dur de la machine d'acquisition et par la suite, étant donné la taille limitée du disque dur, les images sont archivées sur CD-ROM. Un CD-ROM a une capacité de 650Mo ou 700Mo, il peut ainsi contenir 35 ou 38 cas cliniques.

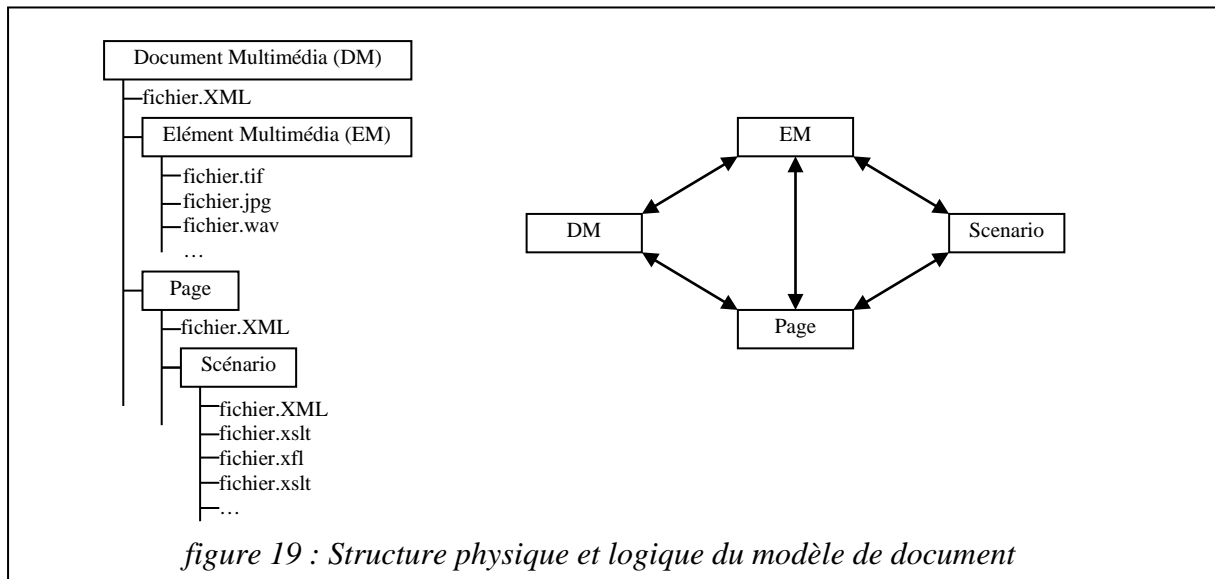
3.4.4. L'intégration des données cliniques sous forme de document

Comme on l'a exprimé dans la première partie, il est indispensable de regrouper les images et les informations cliniques au sein d'un document multimédia. Un modèle de document similaire à celui esquissé dans le projet EMIM est utilisé dans le cadre du système développé (figure 19).

Ce modèle consiste en un agrégat d'informations médicales et techniques regroupées dans une arborescence de fichiers et de répertoires. Un document est constitué d'un répertoire racine, qui contient un fichier XML renfermant les informations communes à l'entièreté du document, un sous-répertoire recelant l'ensemble des éléments multimédia du document et une série de sous-répertoires correspondant chacun à une page du document. Chaque sous-répertoire de cette série possède un fichier XML recouvrant les informations spécifiques à la page et une seconde série de sous-répertoires correspondant aux scénarios pouvant être appliqués à la page afin d'obtenir différentes manifestations du document. Le sous-répertoire de chaque scénario comprend un fichier XML permettant la coordination du scénario et une collection de fichiers de transformation tels que les fichiers XSLT et XFL.

Cette structure physique du modèle de document permet de mettre en évidence les quatre éléments de base : le document, les éléments multimédia, les pages et les scénarios. La dimension logique est facilement mise en œuvre dans ce modèle de document. En effet, les liens logiques entre les pages et le document peuvent être précisés dans le fichier XML global au document, mais apparaissent aussi dans la structure physique du modèle puisque chaque page correspond à un sous-répertoire. Les rapports logiques entre le document et les éléments multimédia sont également exprimés via la structure physique puisque l'ensemble des éléments multimédia est contenu dans un sous-répertoire. Les relations logiques entre les pages et les éléments multimédia peuvent, quant à elles, être déterminées dans le fichier XML de chaque page. Enfin, les dépendances logiques entre les scénarios et les pages apparaissent

aussi dans la structure physique du modèle puisque chaque scénario correspond à un sous-répertoire tandis que les dépendances entre les scénarios et les éléments multimédia peuvent être énoncées dans le fichier XML des scénarios.



Dans ce modèle de document, trois fichiers XML sont utilisés. Une DTD (Document Type Definition) doit être créée pour chacun de ces fichiers, sur base des informations qu'il est susceptible de contenir. Ces informations doivent également être déterminées. Dans le cas d'une analyse cyto-hématologique, l'examen doit être représenté par une page dans ce modèle de document, tandis que les images produites durant l'analyse correspondent à des éléments multimédia.

Ensuite, comme le document peut potentiellement être transmis au serveur « Télémis », vers l'ADICAP ou via E-mail à tout autre expert possédant une adresse, il est nécessaire de concevoir les scénarios adaptés aux différents envois afin d'obtenir les manifestations dont le contenu et le format sont adéquats. Par conception de scénario, on comprend la production d'un fichier XML de coordination respectant la DTD et la génération de feuilles de transformation, sous forme de fichiers XSLT, XFL.

Tel qu'il l'a été énoncé dans la première partie, ces feuilles de transformation doivent formaliser les dimensions spatiale, temporelle et hypermédia, nécessaires à la transformation du document en manifestation, chaque dimension étant associée à une feuille de transformation. Mais ces feuilles doivent également permettre le formatage du document en une manifestation au format adéquat. Cependant, vu que l'expression de la dimension spatiale est simpliste et que les dimensions temporelle et hypermédia sont presque absentes pour les manifestations attendues (galerie d'images), les deux étapes de production des présentations (transformation, formatage) peuvent être regroupées. Dès lors, on ne dispose donc plus d'une feuille de transformation pour chaque dimension. On peut donc se limiter à la création de feuilles de transformation dans lesquelles sont intégrés l'expression des différentes dimensions et le formatage.

En conclusion, l'intégration des données cliniques et des images issues de l'analyse s'effectue via la composition d'un document respectant le modèle décrit ci-dessus. Dans l'optique de l'automatisation, cette composition doit bien sûr pouvoir être automatisée mais il est, en outre, opportun de pouvoir composer un document à partir des images sélectionnées et acquises de façon manuelle.

3.4.5. La transmission des documents

Premièrement, un document, ou plutôt une manifestation du document, doit pouvoir être transmise à partir de la station d'acquisition vers le serveur d'acquisition de « Télémis ». Or la station d'acquisition et le serveur d'acquisition de « Télémis » sont tous deux connectés au réseau local. On peut donc conserver, dans cette manifestation, des images non compressées et à haute résolution.

Cependant, il ne faut pas oublier que le serveur « Télémis » ne permet l'échange d'images que si celles-ci revêtent le format DICOM et uniquement via le protocole DICOM. Dès lors, si les images sont stockées dans un format de fichier différent de DICOM, il faut prévoir un mécanisme de conversion de format de fichier, qui doit être exprimé dans le scénario aboutissant à la manifestation.

Il faut encore remarquer que seule cette transmission, c'est-à-dire celle vers le serveur « Télémis », est automatisée. Cette automatisation intervient dans le but de satisfaire l'objectif formulé précédemment.

Deuxièmement, une manifestation du document doit également pouvoir être transmise vers le serveur de l'ADICAP, soit à partir de la station d'acquisition, soit à partir de n'importe quel ordinateur exploitant le composant « Télémis-Reception ». Ce qui n'est pas le cas puisque « Télémis » ne permet de transmettre un document que vers une autre boîte « Télémis-Pim ». Dans ce second cas de figure, la manifestation du document ne transite plus uniquement par le réseau local mais utilise l'Internet afin d'atteindre le serveur de l'ADICAP. Etant donné le débit étroit qui est réservée à ce trafic, il est plus que nécessaire de diminuer la taille requise par la manifestation, à laquelle les images contribuent pour la majeure partie. Donc le meilleur moyen de réduire la quantité de données est l'utilisation d'images compressées. Puisque les images de cytologie hématologique n'admettent des résultats satisfaisants qu'avec l'emploi d'une compression avec perte et que ce type de compression induit des altérations dans les images lorsque le degré de compression est trop élevé, il convient donc de trouver un compromis entre la qualité des images et leur taille. Il semble qu'un taux de compression de 50 ou un niveau de qualité de compression de 55 fournit un compromis raisonnable, permettant d'obtenir des images de 30 Ko au lieu de 360 Ko pour les originales. [Mercier00] Ce mécanisme de compression doit aussi pouvoir être traduit dans le scénario.

Enfin, une manifestation du document doit également pouvoir être transmise vers un serveur de mail, soit à partir de la station d'acquisition, soit à partir de n'importe quel ordinateur exploitant le composant « Télémis-Reception ». Cette situation rejoint la précédente puisque le serveur de mail peut être localisé en dehors du réseau local.

En ce qui concerne les deux dernières transmissions, une réflexion doit encore être menée à propos du format des images et de la manifestation à transmettre. En effet, il faut s'assurer que les formats utilisés sont accessibles au plus grand nombre de destinataires.

3.4.6. La visualisation, l'édition et l'interprétation des documents

Le système « Télémis » fournit grâce à son module « Télémis-Reception », le moyen privilégié de visualisation des documents. En effet, d'abord les documents peuvent être transmis automatiquement. Ensuite, ce module dispose les images en galerie et procure des outils efficaces de manipulation d'images. Cependant, aucune opération d'édition n'est possible à partir de ce module. De plus le besoin de certaines fonctionnalités supplémentaires lors de la visualisation est énoncé par les spécialistes. Ces fonctionnalités sont : la sélection et

l'affichage de seulement une partie des images de la galerie, le reclassement des images de la galerie et la comparaison de chaque image de la galerie avec une image provenant d'une base de données contenant les cas typiques de cyto-hématologie.

Il faut donc remédier à ces lacunes en développant les nouvelles fonctionnalités dans le module « Télémis-Reception » et en mettant au point un moyen d'éditer les documents, en respectant les principes énoncés dans la première partie.

4. Les technologies utilisées

Dans cette partie, le but est de commenter les technologies utilisées dans le système. Evidemment, toutes les technologies ne sont pas commentées, seules celles qui sont moins connues ou plus complexes sont expliquées dans ce chapitre.

Premièrement, un modèle de document est utilisé dans le cadre du système développé. Ce modèle de document fait appel à différentes technologies telles que XML, XSLT.

Deuxièmement, les transmissions des documents ou plutôt d'une manifestation de ces documents vers le serveur d'acquisition du système « Télémis » doivent être réalisées selon le format DICOM et en utilisant le protocole DICOM.

Ces technologies sont explicitées dans la suite de ce chapitre.

4.1. XML

Le XML (eXtended Markup Language) est un sous-ensemble du SGML. SGML est le père de tous les langages à balises. Le XML n'est pas un langage, mais une syntaxe. Cette syntaxe décrit les règles à suivre pour structurer des informations en utilisant des balises.

Un langage particulier XML est défini par sa DTD (Document Type Definition) qui respecte également la syntaxe XML. Une DTD est donc un document XML définissant un langage particulier à balises respectant la syntaxe XML.

Un document XML peut être valide selon une DTD et si la validation est un succès, le document est dit conforme XML basé sur la DTD particulière. Si le document XML n'obéit à aucune DTD mais qu'il utilise sans erreur la syntaxe XML, il est dit bien formé.

Les documents XML (*figure 20*) se composent d'unités de stockage d'informations appelées éléments. Chaque élément est constitué de caractères, certains formant les données et le reste formant le balisage. XML fournit simplement les mécanismes pour imposer des contraintes sur le balisage. Le balisage décrit une structure logique d'informations arborescentes.

En effet, les éléments ont toujours une balise de début et une balise de fin. Le nom des balises de début et de fin spécifie le type d'élément. Ces mêmes balises peuvent également spécifier des attributs. Ces attributs peuvent être considérés comme fils de l'élément dans la structure logique.

Enfin, l'élément contient entre ses deux balises un contenu. Le contenu est de quatre types, soit ce sont des données textuelles, des instructions de traitements, des commentaires ou enfin un autre élément. Dans ce cas, la balise de début et la balise de fin de ce dernier élément sont contenues entre les balises du premier. Quelque soit le type du contenu d'un élément, on peut toujours considérer ce contenu comme fils de l'élément auquel il appartient.

```

<?xml version="1.0"?>
<document>
  <page>
    <titre>C'est mon article</titre>
    <auteur nom="Georges" prenom="Benoit">bgeorges@info.fundp.ac.be</auteur>
    <content>...</content>
    ...
  </page>
</document>

```

figure 20 : Exemple de document XML

4.2. XSLT

Le langage XSLT permet de définir des règles de transformations destinées à transformer un document XML conforme XML basé sur une DTD particulière en un autre. Cela signifie que chaque document XSLT, appelé feuille de style, regroupe un ensemble de règles de transformations permettant de passer d'une DTD à une autre. Pour cette raison, elles peuvent d'ailleurs être utilisées en chaîne : une transformation A de la DTD1 à la DTD2 et une transformation B de la DTD2 à la DTD3.

Le langage XSLT est lui-même défini avec la syntaxe XML, cela signifie qu'une feuille de style XSLT est un document XML bien formé. Cette réalité permet d'imaginer la transformation via XSLT de feuilles de style XSLT, ce qui correspond à une transformation de transformations.

La différence principale entre les transformations réalisées avec XSLT par rapport aux autres transformations est que pour effectuer une transformation avec XSLT, il suffit de décrire le résultat à obtenir dans une feuille de style, puis fournir la description à un processeur XSLT (figure 21), plutôt que de devoir spécifier comment obtenir le résultat. C'est donc le processeur XSLT qui se charge d'effectuer les transformations proprement dites. Ce processeur XSLT lit la feuille de style puis le document source (XML) et enfin applique les règles de transformations contenues dans la feuille de style afin de produire le document cible.

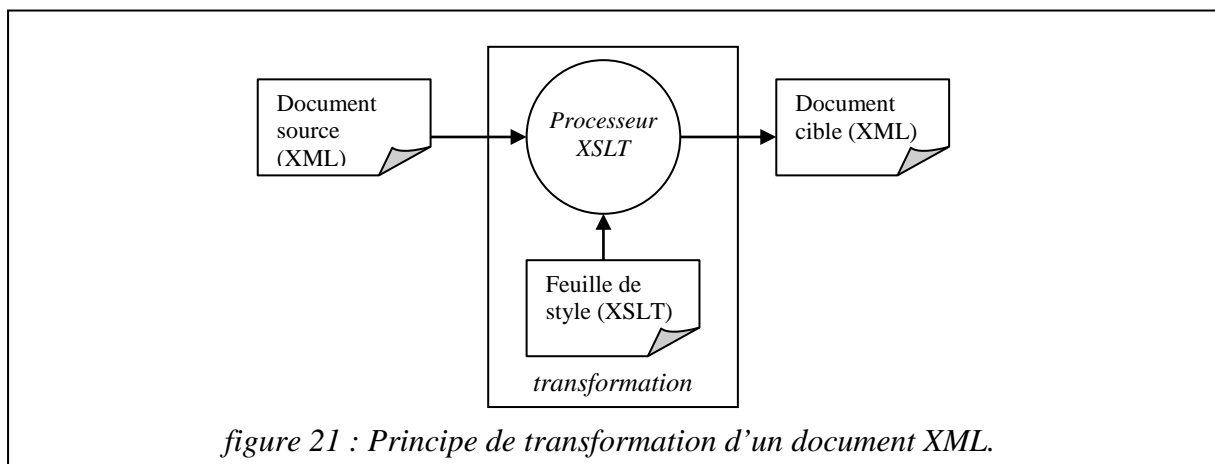
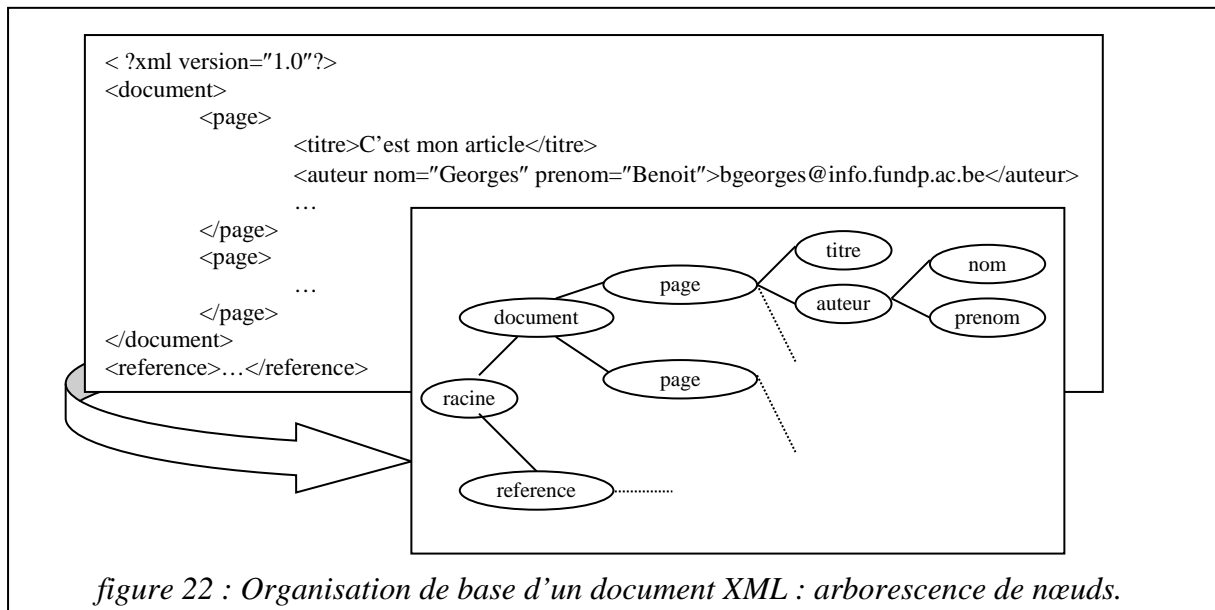


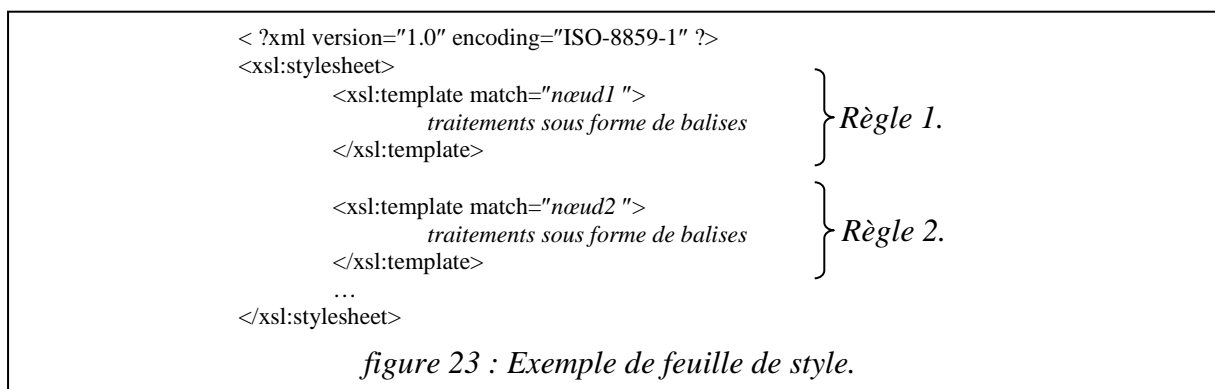
figure 21 : Principe de transformation d'un document XML.

Pour pouvoir appliquer les règles de transformations, un processeur XSLT doit composer avec le principe selon lequel les informations sont organisées dans un document XML. L'organisation de base des informations peut être décrite de manière abstraite comme un ensemble de nœuds assemblés hiérarchiquement sous forme d'arbre (figure 22), chaque nœud contenant une partie des informations.



Le processeur XSLT crée donc une structure logique arborescente à partir du document XML et lui fait subir des transformations selon les règles contenues dans la feuille de style, pour produire un arbre résultat correspondant au document cible. Pour un processeur XSLT, un document XML peut être constitué de sept types de nœuds : racine, élément, texte, attribut, espace de noms, instruction de traitement et commentaire.

Chaque règle de transformations définit donc des traitements à effectuer sur un nœud de l'arbre source. Une règle de transformations est composée de balises dont la première « `<<xsl:template>>` » et la dernière « `</xsl:template>>` ». La première balise est complétée par un attribut « `match` » dont la valeur permet de définir le ou les nœuds du document XML sur lesquels s'applique la transformation :



La composition d'une règle de transformations est réalisée à l'aide de balises qui permettent de sélectionner et d'effectuer des opérations sur les éléments du document XML. Leur syntaxe est la suivante : « `<<xsl:nom [attributs]/>>` ». Voici une liste des balises de transformations les plus couramment utilisées :

Nom	Attributs	Signification
apply-templates	select mode	Permet d'appliquer les règles correspondant à l'attribut <i>mode</i> sur les nœuds sélectionnés par l'attribut <i>select</i> .
call-template	name	Permet d'invoquer une règle dont le nom est <i>name</i> .
with-param	name	Permet de définir un paramètre dont le nom est <i>name</i> lors d'un appel de règle xsl:call-template et xsl:apply-template . La règle appelée doit déclarer ce paramètre à l'aide de xsl:param .
param	name	Permet de définir un paramètre dont le nom est <i>name</i> .
variable	name	Permet de définir une variable dont le nom est <i>name</i> .

choose		Permet de choisir entre plusieurs alternatives de traitement. Cette balise est suivie de xsl:when ou xsl:otherwise qui permettent de séparer les alternatives en fonction de certaines conditions.
if	test	Permet de choisir si un traitement doit être effectué en fonction d'une condition spécifiée par l'attribut <i>test</i> .
for-each	select	Permet l'itération d'un traitement pour tous les nœuds sélectionnés par l'attribut <i>select</i> .
value-of	select	Permet d'évaluer, comme un string, la valeur du nœud sélectionné par l'attribut <i>select</i> .
copy		Permet de copier le nœud courant dans le document cible.
copy-of	select	Permet de copier la valeur des nœuds sélectionnés par l'attribut <i>select</i> dans le document cible.
element	name	Permet de créer un nouvel élément dont le nom est <i>name</i> dans le document cible.
attribute	name	Permet d'ajouter un attribut dont le nom est <i>name</i> à un élément.
text		Permet d'écrire un texte dans le document cible.
document	href	Permet de diriger le résultat de la transformation vers un autre document cible dont l'URL est spécifiée par l'attribut <i>href</i> . Un document source peut donc être à l'origine de plusieurs documents cibles.
include	href	Permet d'inclure les règles de transformations d'une autre feuille de style dont l'URL est spécifiée par l'attribut <i>href</i> .

figure 24 : Liste des balises principales de transformations XSLT.

La composition d'une règle de transformations nécessite donc un moyen d'expression de sélection de nœuds, pouvant être utilisé afin de fournir une valeur aux attributs *select* des différentes balises de transformations. De plus, certaines balises requièrent la possibilité de formuler des conditions. Ces exigences sont satisfaites par XPath qui définit une syntaxe fournissant les outils (un ensemble de types de données utilisées pour représenter les informations et les nœuds ; et un ensemble de fonctions pour manipuler ces types de valeurs) pour localiser les nœuds et les informations contenues dans ces nœuds, à partir de la structure abstraite. L'ensemble des types de valeurs inclut les strings, les entiers, les valeurs booléennes et des expressions de nœuds telles que : le nœud fils, le nœud père, l'attribut du nœud courant, etc. La syntaxe des expressions de nœud ressemble à celle utilisée dans les systèmes pour adresser un répertoire.

Expression-Nœud	Signification	
.	nœud courant	
/	nœud racine	
/ chemin-relatif	nœuds de chemin relatif par rapport au nœud racine	
<i>expression-nœud1</i> <i>expression-nœud2</i>	union de deux ensembles de nœud	
<i>expression-nœud1</i> [<i>predicat</i>]	sous-ensemble de l'ensemble de nœuds <i>expression-nœud1</i> qui respecte le <i>predicat</i> (prédictat exprimé grâce à XPath)	
<i>expression-nœud1</i> / <i>chemin-relatif</i>	nœuds de chemin relatif par rapport aux nœuds de l'ensemble <i>expression-nœud1</i>	
	<i>chemin-relatif</i>	Signification
	*	tous les nœuds fils
	<i>name</i>	tous les nœuds fils dont le nom est <i>name</i>
	@*	tous les attributs
	@ <i>name</i>	tous les attributs dont le nom est <i>name</i>
	..	tous les nœuds parents

figure 25 : Syntaxe (incomplète) des expressions de nœuds de XPath.

4.3. DICOM

4.3.1. Le Standard DICOM

En 1983, l'« American College of Radiology » (ACR) et la « National Electrical Manufacturers Association » (Nema) se sont associés, pour développer un standard permettant de promouvoir la communication d'images digitales, de faciliter le développement et l'expansion de systèmes d'archivage et de communication d'images qui puissent s'intégrer avec les autres systèmes d'informations hospitaliers. Cette coopération a abouti au standard « Digital Imaging and Communications in Medicine » (DICOM).

La spécification du standard DICOM est réalisée en 10 parties :

La partie 1 (Introduction and Overview) fournit une introduction et une vue d'ensemble décrivant la structure du standard et des autres parties.

La partie 2 (Conformance) spécifie les exigences générales qui doivent être respectées par les implémentations revendiquant la conformité à la norme DICOM.

La partie 3 (Information Object Definitions) spécifie la structure et les attributs des objets contenant l'information et qui doivent subir des opérations selon les « Service Classes » spécifiés. Ces objets incluent les images, les informations spécifiques au patient, etc.

La partie 4 (Service Class) définit les opérations qui peuvent être appliquées aux instances d'« Information Object Definitions » afin de fournir un service particulier. Ces services incluent l'enregistrement, la suppression de certaines informations, etc. Cette partie définit donc les différentes combinaisons des « Information Object Definitions » avec les « Service Classes », ce qui détermine les « Service Object Pair Classes ».

La partie 5 (Data Structure and Semantics) spécifie l'encodage des objets contenant l'information sous forme de contenu de messages qui sont échangés entre les entités d'application.

La partie 6 (Data Dictionary) définit l'information spécifique contenue dans chaque attribut des objets.

La partie 7 (Message Exchange) spécifie l'encodage des opérations et les protocoles utilisés pour échanger les messages en s'appuyant sur les supports de communication définis dans les parties 8 et 9. Ces opérations sont utilisées pour accomplir les services définis par les « Service Classes ».

La partie 8 (Network Communication Support for Message Exchange) définit les services et protocoles utilisés pour échanger les messages directement sur les réseaux OSI ou TCP/IP.

La partie 9 (Point-to-Point Communication Support for Message Exchange) définit les services et les protocoles utilisés pour échanger les messages entre deux machines contiguës sans utiliser des protocoles existants.

La partie 10 (Media Storage and File Format for Media Interchange) spécifie un format de fichier permettant de regrouper les informations pouvant être échangées via DICOM, c'est-à-dire l'ensemble des informations contenues dans les « Information Object » et les informations concernant les services à rendre sur ces informations « Service Class ».

Donc on peut conclure que pour faciliter l'interopérabilité des équipements d'imagerie médicale, le standard DICOM spécifie un ensemble de protocoles à suivre par les appareils voulant communiquer, un ensemble de modèles permettant l'organisation logique et physique de données à échanger et un ensemble d'opérations ou de services à fournir pour ces données. Enfin, DICOM définit un format de fichier afin de pouvoir stocker les données sur un disque.

L'organisation logique des données est détaillée dans des modèles entité-association qui sont

utilisés afin de fournir une définition abstraite des éléments d'information et des relations qui les animent. Ces modèles de représentation sont utiles lorsque l'on considère l'aspect communicatif, ils permettent de savoir facilement quels types d'objets vont être transmis. Le standard DICOM est orienté objet. Des objets, appelés « Information Object » (IO), sont donc basés sur les modèles de représentation. On considère donc les modèles qui les définissent comme leur « Information Object Definition » (IOD). Un IOD représente donc une classe d'objets partageant les mêmes propriétés ou informations. Cependant, ces classes peuvent être altérées, des informations peuvent être omises ou ajoutées. *La norme DICOM est orientée objet, cela signifie également que chaque objet DICOM contient à la fois les informations, appelées « Information Object » (IO), mais aussi les méthodes, appelées « Service » (S), que doivent subir ces informations [Derhy], c'est-à-dire ce qu'il faut faire avec les informations. La combinaison d'un IO avec des S est appelée « Service Object Pair » (SOP). Pour se conformer à une classe de SOP, une entité applicative ou « Application Entity » (AE) doit pouvoir gérer un type particulier d'informations et réaliser un type spécifique de traitements correspondant à la définition de cette classe de SOP. Par ailleurs la classe de SOP doit spécifier si le service est employé en tant qu'utilisateur, « Service Class User » (SCU), ou en tant que fournisseur, « Service Class Provider » (SCP). Lorsqu'une AE possède un objet DICOM, si elle est en conformité de classe SCU alors elle s'adresse à une AE qui fournit la classe SCP correspondante. Il en résulte alors une phase de négociation pendant laquelle les AE se mettent d'accord sur le type de données à échanger et la façon dont elles vont les échanger [Derhy].*

L'organisation physique (figure 26) des informations s'effectue sous forme séquentielle, en « Data Set ». Un « Data Set » est une instance de classe de SOP. Il est composé d'informations élémentaires « Data Element » correspondant aux attributs des IO et aux services. *Une information élémentaire peut avoir trois structures. Deux d'entre elles contiennent une représentation de valeurs mais diffèrent dans la façon dont leur longueur est exprimée. Toutes les trois structures contiennent trois champs : un Tag, la longueur de la valeur et la valeur [Nema99].*

Le premier champ (Tag) est codé sur 16bits (4 bytes), il s'agit d'une balise, répertoriée dans le dictionnaire DICOM, qui indique le type d'information qui suit. Cette balise est décomposée en deux parties, une signalant le groupe (2 bytes) auquel appartient l'information et l'autre indiquant l'élément du groupe (2 bytes) correspondant à l'information. La balise permet d'identifier l'information élémentaire. Le deuxième champ (longueur) est codé sur 16 ou 32 bits (2 ou 4 bytes) selon la structure et spécifie la longueur du troisième champ (Valeur).

Le quatrième champ (Représentation de valeur), unique à deux structures, est codé sur 2 bytes et spécifie le type de données du troisième champ.

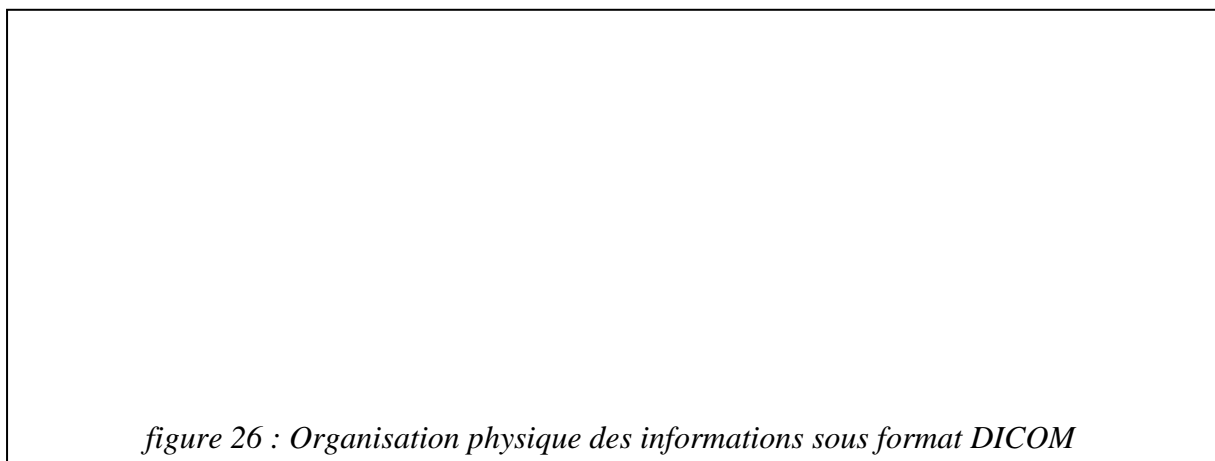
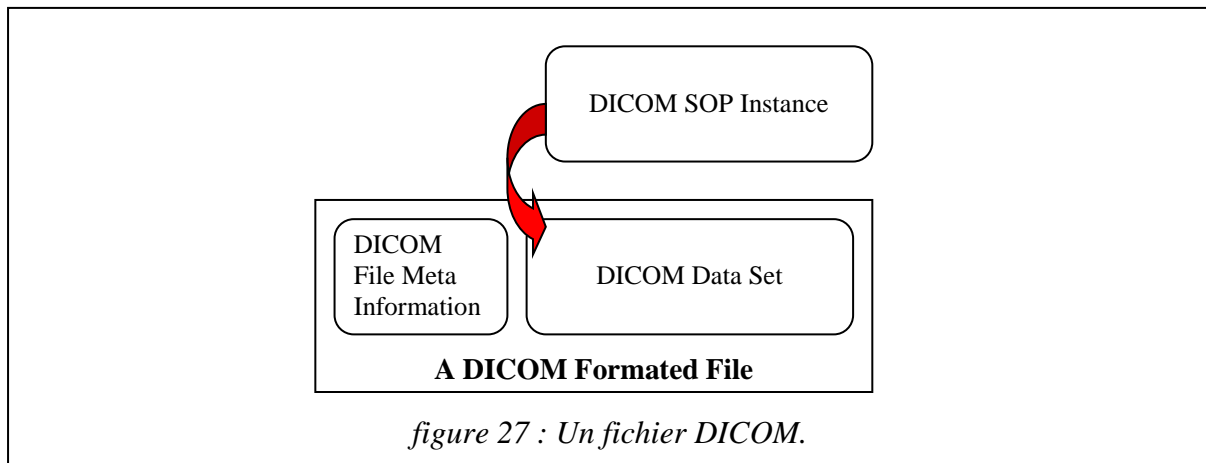


figure 26 : Organisation physique des informations sous format DICOM

Enfin, le format de fichier DICOM (figure 27) permet d'encapsuler dans un fichier un Data Set, représentant une instance de classe de SOP. Chaque fichier ne contient qu'un seul Data Set qui est placé dans le fichier à la suite d'une en-tête (File Meta Information FMI) [Nema99].

L'en-tête est composée d'un préambule de 128 bytes, d'un préfixe de 4 bytes, « DICM », et d'un ensemble d'informations décrivant le Data Set qui suit.



4.3.2. DICOM et la microscopie

Le standard DICOM permet de définir des classes d'objets, « Information Object Definition ». D'ailleurs, il existe une classe spécifique réservée à la représentation des informations en provenance d'une station de microscopie, « VL Microscopic Image IOD ». On peut retrouver l'ensemble des informations élémentaires pour le « VL Microscopic Image IOD » à l'annexe A.32.2 de la partie 3 de la spécification du standard DICOM.

Le tableau qui suit décrit un fichier DICOM reprenant les informations du « File Meta Information » et les informations élémentaires obligatoires pour le « VL Microscopic Image Storage SOP » :

TAG	NOM	Type	VR	Multiplicité - Valeur
<i>Header</i>				
	File Preamble	1		(128 bytes)
	DICOM Prefix	1		1 - DICM (4 bytes)
(0002,0000)	Group Length	1	UL	1 - ?
(0002,0001)	File Meta Information Version	1	OB	1 - ?
(0002,0002)	Media Storage SOP Class UID	1	UI	1 - 1.2.840.10008.5.1.4.1.1.77.1.2
(0002,0003)	Media Storage SOP Instance UID	1	UI	1 - ?
(0002,0010)	Transfer Syntax UID	1	UI	1 - 1.2.840.10008.1.2 [.1 or .2]
(0002,0012)	Implementation Class UID	1	UI	1 - ?
<i>Data Set</i> <i>Voir PS 3.3 (Annexe A, C)</i>				
(0008,0008)	Image Type	1	CS	2 – ORIGINAL, PRIMARY
(0008,0016)	SOP Class UID	1	UI	1 - 1.2.840.10008.5.1.4.1.1.77.1.2
(0008,0018)	SOP Instance UID	1	UI	1 - same (0002,0003)
(0008,0020)	Study Date	2	DA	1 - ?
(0008,0030)	Study Time	2	TM	1 - ?
(0008,0050)	Accession Number	2	SH	1 - ?
(0008,0060)	Modality	1	CS	1 - GM
(0008,0070)	Manufacturer	2	LO	1 - ?
(0008,0090)	Referring Physician's Name	2	PN	1 - ?
(0010,0010)	Patient's Name	2	PN	1 - ?
(0010,0020)	Patient ID	2	LO	1 - ?

(0010,0030)	Patient's BirthDate	2	DA	1 - ?
(0010,0040)	Patient's sex	2	CS	1 - [M or F or O]
(0020,000D)	Study Instance UID	1	UI	1 - ?
(0020,000E)	Series Instance UID	1	UI	1 - ?
(0020,0010)	Study ID	2	SH	1 - ?
(0020,0011)	Series Number	2	IS	1 - ?
(0020,0013)	Instance Number	2	IS	1 - ?
(0028,0002)	Samples per Pixel	1	US	1 - 3
(0028,0004)	Photometric Interpretation	1	CS	1 - RGB
(0028,0006)	Planar Configuration	1C	US	1 - [000 or 001]
(0028,0010)	Rows	1	US	1 - ?
(0028,0011)	Columns	1	US	1 - ?
(0028,0034)	Pixel Aspect Ratio	1C	IS	2 - ?
(0028,0100)	Bits Allocated	1	US	1 - 8
(0028,0101)	Bits Stored	1	US	1 - 8
(0028,0102)	High Bit	1	US	1 - 7
(0028,0103)	Pixel Representation	1	US	1 - 0000
(0028,2110)	Lossy Image Compression	2	CS	1 - 00
(0040,050A)	Specimen Accession Number	1	LO	1 - ?
(0040,0550)	Specimen Sequence	2	SQ	1 - Vide
(7FE0,0010)	Pixel Data	1	OW or OB	1 - L'image

figure 28 : Ensemble des informations du « File Meta Information » et des informations élémentaires obligatoires pour le « VL Microscopic Image Storage SOP »

4.3.3. La compatibilité entre DICOM et TIFF

Bien que la partie 10 du standard DICOM, traitant du format fichier, soit abondamment employée dans le domaine médical et qu'elle recouvre de nombreux avantages tels que la facilité d'interopérabilité, de communication, d'archivage et l'adjonction des autres informations médicales (patient, ...), ce format souffre d'un inconvénient non négligeable : beaucoup d'outils d'imagerie logiciels ne sont pas capables de traiter ce type de format. Pour pouvoir éviter cet inconvénient, on peut dès lors penser à intégrer le format de fichier de la partie 10 du standard DICOM avec un format plus usuel afin d'obtenir un format de fichier compatible avec les deux formats. Il semble que le format « Tag-based Image File Format » (TIFF) soit le plus adéquat pour effectuer une telle intégration. Une tentative d'élaboration d'un format compatible avec TIFF et le format de fichier de la partie 10 du standard DICOM est d'ailleurs présentée à l'annexe 8 (VOIR p. 89).

Troisième partie : Résultats

5. La conception du modèle de document

A partir des DTD déjà développées dans le cadre du projet EMIM et des spécifications DICOM et ADICAP, on peut déterminer un ensemble d'informations qui pourraient être impliquées dans un document multimédia à caractère médical, ainsi que les relations logiques qui les unissent.

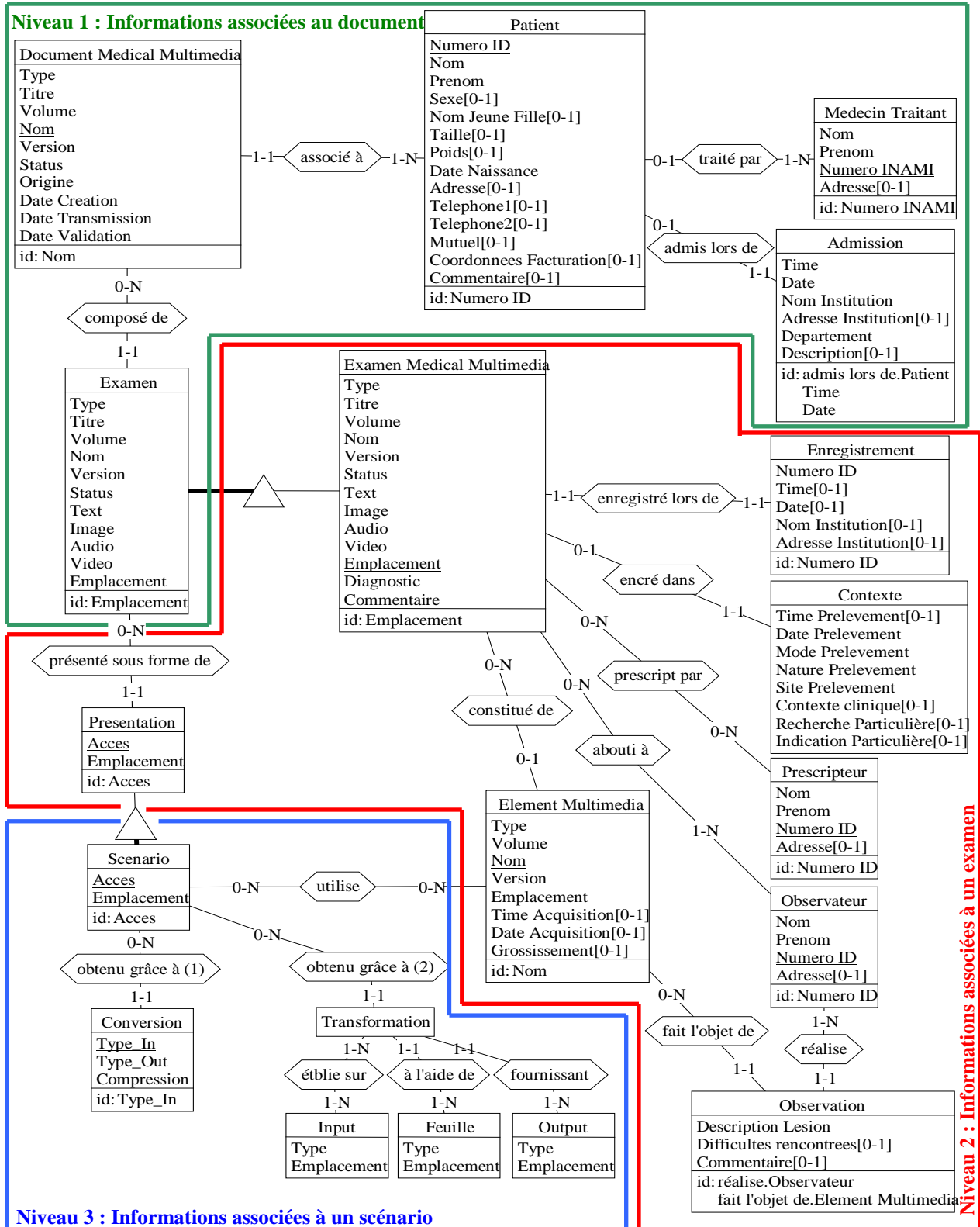


figure 29 : Schéma E-A du modèle de document multimédia médical

acquises dans un format TIFF doivent être converties en format JPEG lorsqu'elles sont transmises au serveur ADICAP. Pour les images, on a également indiqué le besoin d'un mécanisme de compression. L'entité « Conversion » permet de mettre en œuvre ces deux processus. L'attribut « Type_In » définit le type des éléments multimédia qui doivent être convertis et l'attribut « Type_Out » permet de spécifier le type vers lequel les éléments multimédia doivent être convertis. Ces deux attributs prennent comme valeur des types MIME : text/html, image/gif, image/jpeg, image/tiff, image/dcm, etc. L'attribut « Compression » permet de définir la compression pour les éléments multimédia convertis. La valeur de cet attribut dépend évidemment du type des éléments multimédia convertis. Certains types d'éléments multimédia ne supportent aucune compression, d'autres supportent une compression sans variation possible de celle-ci et d'autres encore permettent une compression qui peut être paramétrée par un taux. Donc les valeurs acceptées sont soit « no » si aucune compression n'est désirée, soit « yes » dans le cas où une compression est désirée, soit un nombre de 0 à 100 pour définir un taux de compression.

Enfin, un scénario est associé à des transformations. Comme on l'a fait remarquer, une transformation s'effectue à l'aide d'une feuille de style appliquée à un ou plusieurs documents sources pour fournir un document cible (*figure 21*). Ces différents concepts sont représentés par les entités : « Transformation », « Input », « Feuille » et « Output ». Lors d'une transformation, il est possible que plusieurs documents sources soient nécessaires mais un seul doit être fourni au processeur pour effectuer correctement la transformation. L'attribut « Type » de l'entité « Input » permet de reconnaître parmi les documents sources, le document primaire car la valeur de son attribut « Type » est « Primaire » tandis que celle des autres documents est « Secondaire ». En outre, il faut connaître le type de transformation qu'il faut effectuer afin de choisir le processeur adéquat (le processeur XFL n'est pas identique au processeur XSLT). L'attribut « Type » de l'entité « Feuille » fournit le moyen de reconnaître le type de transformation. Par exemple, s'il s'agit d'une transformation XSLT, la valeur de cet attribut est « XSLT ». Pour conclure, les attributs « Emplacement » des entités « Input », « Feuille » et « Output » sont destinés à fournir l'emplacement physique des différents fichiers nécessaires pour la transformation.

Lors de l'analyse de l'intégration des données cliniques sous forme de document, on a également énoncé la nécessité de concevoir trois scénarios permettant d'obtenir des manifestations adaptées à un examen de cytologie hématologique et dédiées à la transmission vers « Télémis », l'ADICAP ou via E-mail (VOIR annexe 10, p. 95).

6. La conception du système

6.1. La découpe en composant

Avant tout, le système doit offrir la possibilité de sélectionner, acquérir et stocker les images provenant du microscope de manière manuelle. Le logiciel « Analysis » remplit déjà ces fonctions. Donc aucun développement n'est nécessaire.

Ensuite, le système doit être capable de composer des documents à partir des images obtenues de façon manuelle. Or cette facilité n'est offerte ni par « Analysis », ni par « Télémis ». Il faut donc développer un module de composition de document respectant le modèle décrit. Ce module doit pouvoir être intégré aussi bien à « Analysis » qu'à « Télémis ». En outre, le système doit permettre la transmission d'une manifestation d'un document composé. Pour ce faire, il doit appliquer les scénarios, c'est-à-dire effectuer les transformations nécessaires mais également effectuer la transmission proprement-dite de la

manifestation. Dès lors, un module de transformation et un module de transmission doivent être développés. Bien entendu, ces modules doivent pouvoir être intégrés à « Analysis » et à « Télémis ».

Il faut également développer un module d'édition afin de pouvoir éditer les documents. Une nouvelle fois, ce module doit pouvoir être intégré à « Analysis » et à « Télémis ».

Etant donné l'objectif d'automatisation des étapes de la sélection des images jusqu'à la transmission d'une manifestation du document composé vers le serveur d'acquisition de « Télémis », il est nécessaire de développer un module d'automatisation qui utilise par ailleurs les modules de composition, de transformation et de transmission.

Enfin, on doit développer deux modules d'interfaces, un pour « Analysis » et un pour « Télémis », afin d'intégrer les différentes fonctionnalités disponibles via les autres modules dans l'interface graphique des deux applications. Un module de gestion des demandes d'analyses doit également être élaboré afin de gérer les informations relatives au patient lors d'une analyse automatique.

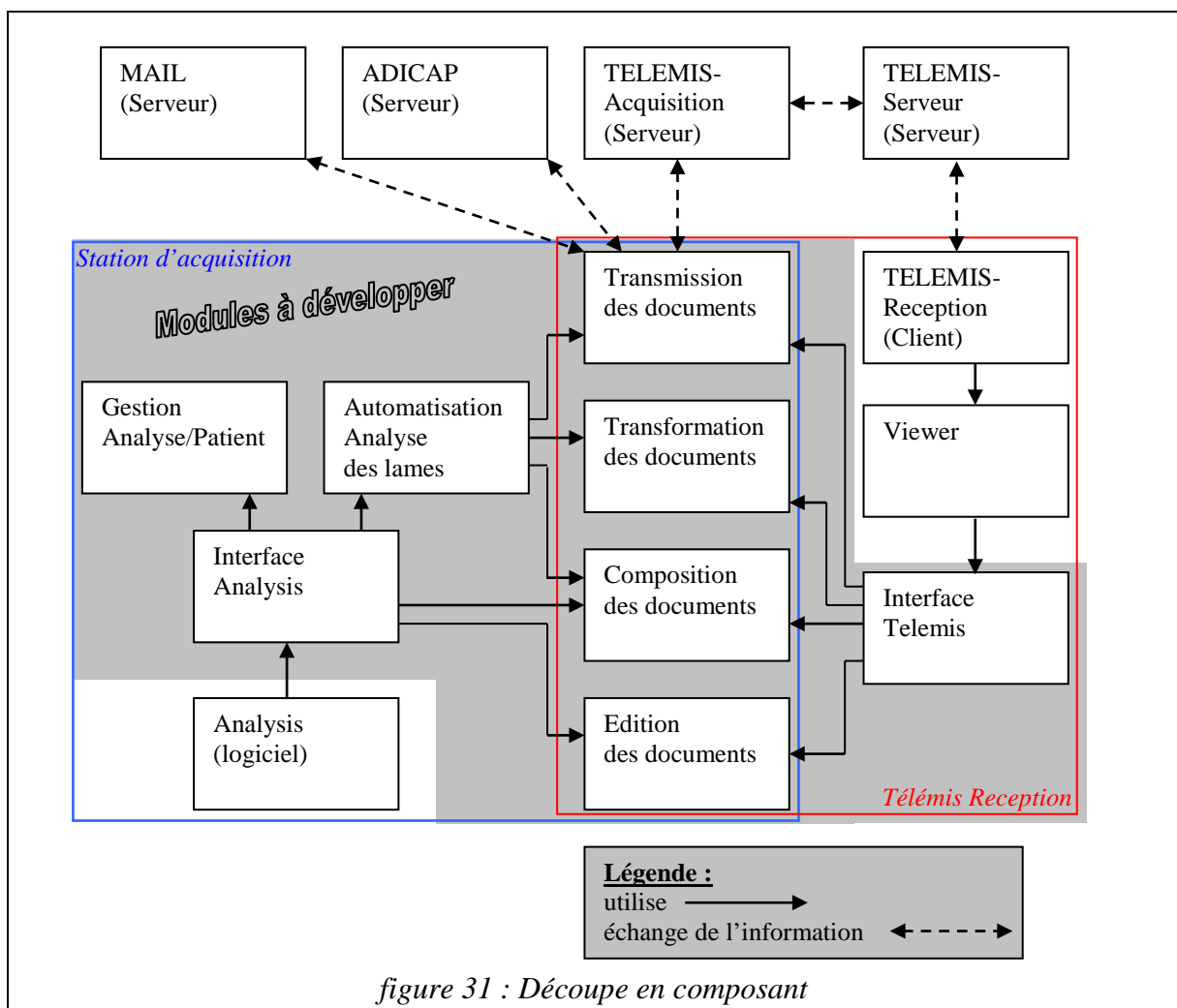


figure 31 : Découpe en composant

L'utilisation du langage de programmation « Java », pour les modules relatifs à la composition, l'édition, la transformation et la transmission des documents est encouragée par le fait que le projet EMIM évolue dans ce cadre technologique. Et il n'est pas impossible que les composants développés dans le projet de télémicroscopie puissent être, en partie, utilisés par l'équipe de recherche travaillant sur le projet EMIM. L'usage multiplateforme de ce langage et les performances dont il fait preuve dans le domaine du multimédia, motivent également son emploi.

Par contre, il semble que pour les modules « Interface Analysis », « Gestion Analyse/Patient » et « Automatisation Analyse des lames », la solution retenue soit celle de modules « Imaging-C » directement intégrés au logiciel « Analysis ». En effet, cette solution semble la plus efficace puisque les fonctions nécessaires à la sélection, l'acquisition et le stockage des images sont déjà disponibles dans le logiciel « Analysis ».

6.2. Le fonctionnement du système

6.2.1. La station d'acquisition

Grâce au module « Interface Analysis », on a pu ajouter à l'interface générale du logiciel « Analysis » une barre de boutons contenant quatre boutons, chacun dédié à une fonctionnalité : le premier est dédié à la composition d'un document, le second à la visualisation, l'édition et la transmission d'un document, le troisième à l'analyse automatique de lames et le quatrième au choix de certains paramètres.

La composition d'un document multimédia médical

Le premier bouton ajouté à l'interface d'« Analysis » (VOIR annexe 11, p. 109), permet d'accéder à la composition d'un document (respectant la description faite dans la partie 4.3.4) avec une seule page renfermant les images précédemment sélectionnées, acquises dans les buffers d'« Analysis ».

Grâce à ce bouton, l'utilisateur obtient une boîte de dialogue proposant de confirmer la volonté de composer un nouveau document mais également de choisir le répertoire dans lequel ce document est composé. S'il révoque sa volonté de composer un document, la boîte de dialogue est simplement fermée. S'il confirme la composition, les images sont stockées dans des fichiers au format TIFF non compressés, placés dans un répertoire temporaire et la composition proprement-dite peut commencer.

Etant donné que les modules « Interface Analysis » et « Composition des documents » ont été réalisés dans deux langages différents, il a fallu lancer, à partir du module « Interface Analysis », un "shell" auquel est associé un fichier « ComposerForAnalysis.bat » qu'il doit interpréter (*figure 32*). Des arguments qui doivent être associés à ce fichier, en l'occurrence le répertoire qui doit contenir le nouveau document et le répertoire temporaire où se trouvent les images, qu'il doit associer à ce fichier sont également joints à ce "shell".

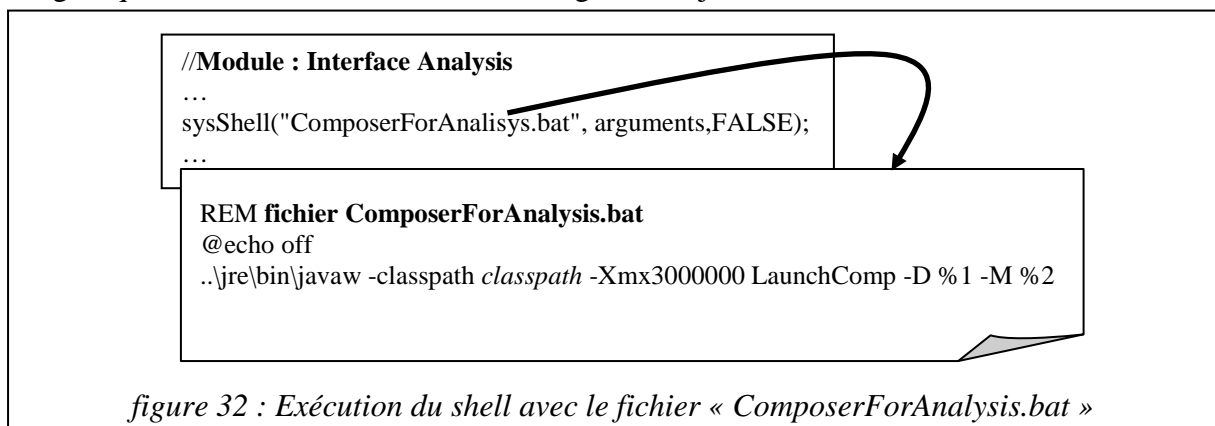


figure 32 : Exécution du shell avec le fichier « ComposerForAnalysis.bat »

Le fichier « ComposerForAnalysis.bat » permet simplement d'activer la machine virtuelle Java sur la classe « LaunchComp.java » (VOIR annexe 16, p. 118) qui utilise évidemment les deux arguments. Cette classe permet d'afficher l'interface graphique suivante, consacrée à l'encodage des différentes informations associées au document et à la page. L'utilisateur a

ensuite l'opportunité d'annuler la composition ou de la confirmer définitivement. Dans ce dernier cas, la composition du document est réalisée grâce à la classe « Composition.java », en deux étapes.

La première étape consiste à composer un document sans page. Le système va donc construire une instance de la classe « DMM.java » (VOIR annexe 15, p. 117) renfermant l'ensemble des informations associées au document qui ont été précédemment encodées. Ensuite il crée, sur le disque local, le répertoire qui doit contenir le nouveau document (*Création1, figure 33*) et dans ce répertoire, un fichier « DMM.xml » (*Création2, figure 33*) dans lequel il écrit les informations retenues dans l'instance de la classe « DMM.java », en respectant la DTD spécifique à ces informations.

La seconde étape consiste à ajouter la page au document. Le système construit une instance de la classe « EMMCH.java » (VOIR annexe 15, p. 117) renfermant l'ensemble des informations associées à la page, qui ont été précédemment encodées. Ensuite, dans le répertoire racine du document, il crée un sous répertoire pour la page (*Création3, figure 33*) et un sous répertoire « EM » (*Création4, figure 33*) qui contiendra les images. Les opérations suivantes consistent, pour chaque image contenue dans le répertoire temporaire, à déterminer les informations qui la concernent et les ajouter à l'instance de la classe « EMMCH.java ». Ensuite, le système doit également copier les images dans le répertoire « EM » (*Copie5, figure 33*). En ce qui concerne les scénarios à associer à la page, le principe est d'aller chercher des scénarios pré-définis et stockés dans un répertoire, une sorte de réservoir à scénarios. Il faut alors que l'ensemble des fichiers composant ce scénario soit copié dans un sous-répertoire du répertoire de la page (*Copie6, figure 33*). Puisque les scénarios ont changé d'emplacement, le système met à jour les informations ayant trait aux emplacements des inputs, outputs et feuilles de style dans ces scénarios. Ensuite, il met à jour l'instance de la classe « EMMCH.java » afin d'indiquer la présence de nouveaux scénarios.

Les dernières opérations consistent à créer dans le répertoire de la page, un fichier « EMMCH.xml » (*Création7, figure 33*) dans lequel sont écrites les informations retenues dans l'instance « EMMCH.java », en respectant la DTD spécifique à ces informations. Enfin, le fichier « DMM.xml » est mis à jour afin de prendre en compte la nouvelle page ajoutée (*Mises à jour8, figure 33*).

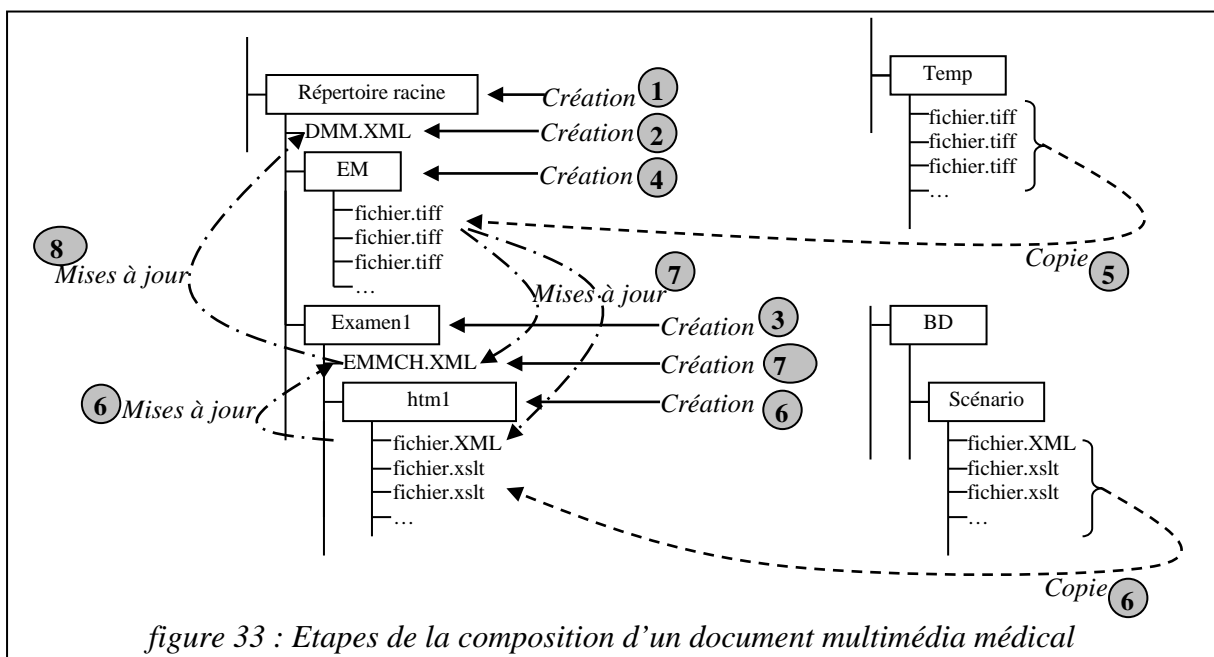


figure 33 : Etapes de la composition d'un document multimédia médical

La visualisation, l'édition et la transmission d'un document multimédia médical

Le deuxième bouton ajouté à l'interface d'« Analysis » (VOIR annexe 12, p. 110), permet d'accéder à la visualisation, l'édition et la transmission d'un document bien formaté présent sur un disque local.

Grâce à ce bouton, l'utilisateur obtient une boîte de dialogue permettant de sélectionner le document que l'on veut éditer ou transmettre. Cette boîte de dialogue présente une liste dont les entrées, sauf la première, permettent d'accéder aux documents créés lors de la dernière analyse automatique. La première entrée permet d'accéder à une seconde boîte de dialogue fournissant le moyen de sélectionner, dans le système local de fichiers, le répertoire correspondant à un document déjà créé. Aussitôt qu'un document est sélectionné, l'utilisateur peut accéder à ce document. Etant donné que les modules « Interface Analysis » et « Edition des documents » ont été réalisés dans deux langages différents, il a fallu utiliser le même procédé que pour la composition d'un document. Le système lance, à partir du module « Interface Analysis », un "shell" auquel est joint un fichier « EditerForAnalysis.bat » et un argument, le répertoire qui doit contenir le document (*figure 34*). Ce "shell" doit interpréter le fichier qui lui est joint.

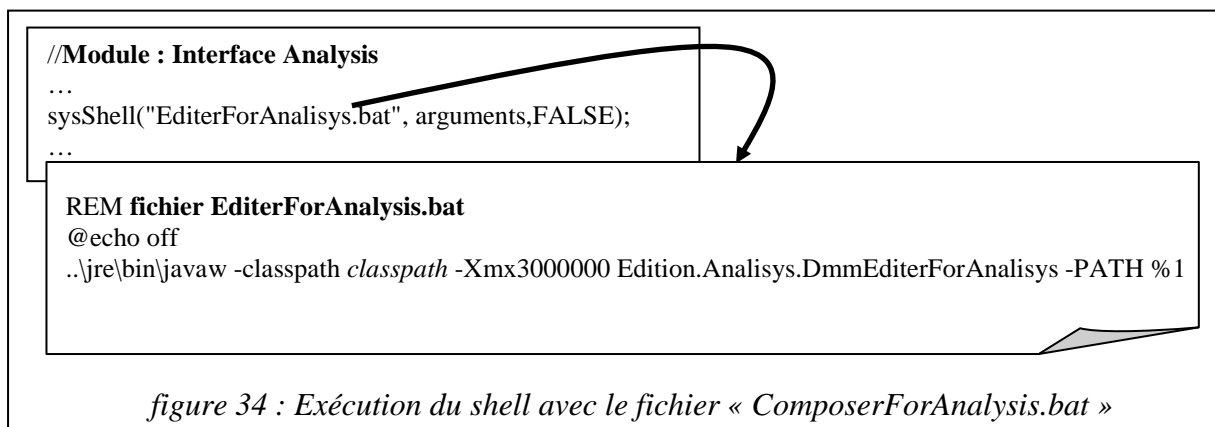


figure 34 : Exécution du shell avec le fichier « ComposerForAnalysis.bat »

Le fichier « EditerForAnalysis.bat » permet simplement d'activer la machine virtuelle Java sur la classe « DmmEditerForAnalisys.java » (VOIR annexe 17, p. 123) qui utilise évidemment l'argument. Cette classe va fournir une interface graphique permettant la visualisation directe du document sélectionné (ou plutôt d'une manifestation d'une page de ce document) mais également son édition et la transmission d'une manifestation d'une page de ce document vers l'ADICAP, vers « Télémis » ou via E-mail.

La visualisation d'un document multimédia médical (VOIR annexe 17, p. 124)

Puisque un document multimédia médical est composé de pages et que chaque page est associée à des scénarios, la visualisation du document consiste à présenter, une à une, les manifestations résultant de ces scénarios. Etant donné que l'ensemble des scénarios prévus dans le cadre du projet aboutissent à des manifestations au format HTML, un visualisateur adapté à ce format, « DMMViewer.java », a été créé.

La visualisation d'un document se déroule en deux étapes.

La première étape consiste à lire les différents fichiers XML du document. Le système va donc lire le fichier « DMM.XML » dans le répertoire racine du document pour obtenir une instance de la classe « DMM.java » contenant les informations du fichier, dont les emplacements des fichiers XML des différentes pages font partie. Ensuite, il suffit que l'utilisateur choisisse la page qu'il désire afficher et le système lit le fichier XML à l'emplacement correspondant afin d'obtenir une instance de la classe « EMMCH.java ». Cette

instance contient évidemment les informations relatives à la page dont les emplacements des fichiers XML des différents scénarios font partie. Enfin, il suffit de lire tous les fichiers XML afin d'obtenir un vecteur d'instances de la classe « Scenario.java ».

La deuxième étape consiste à afficher la manifestation résultant de l'exécution d'un scénario choisi sur base des informations contenues dans l'instance correspondant à ce scénario.

Pour exécuter un scénario, il faut exécuter d'une part l'ensemble des transformations et d'autre part l'ensemble des conversions (figure 35).

L'exécution des transformations est réalisée grâce à la classe « Xtransformation.java ». Tout d'abord, un répertoire, nommé « Manifestation », est créé dans le répertoire du scénario (Création1, figure 35). Ensuite, le système choisit, pour chaque transformation (dans l'ordre d'apparition dans la liste), le processeur qui convient, en fonction du type de la feuille de style associée à la transformation. Une fois le processeur sélectionné, le système lui passe en argument la feuille de style et le document source (Input) primaire pour que le processeur fournisse un document cible (Output). Ce document cible est placé dans le répertoire « Manifestation » (Transformation2, figure 35).

L'exécution des conversions est réalisée grâce à la classe « Media_Conversion.java ».

Le système crée un répertoire, nommé « EM », dans le répertoire « Manifestation » (Création3, figure 35). Ensuite, pour chaque média associé au scénario, il consulte la liste des conversions afin de vérifier s'il y a une conversion à appliquer à ce média. Une conversion est appliquée si le type du média est identique au type d'entrée (attribut « Type_In ») de cette conversion. S'il y a effectivement une conversion à effectuer sur un média, alors il choisit un convertisseur, en fonction du type d'entrée (attribut « Type_In ») et du type de sortie (attribut « Type_Out »), qui est utilisé pour obtenir un média converti qui se place dans le répertoire "EM" précédemment créé (Conversion4, figure 35). Par contre, si aucune conversion ne doit être appliquée pour un média, alors ce média est tout simplement copié dans le répertoire "EM" (Copie5, figure 35).

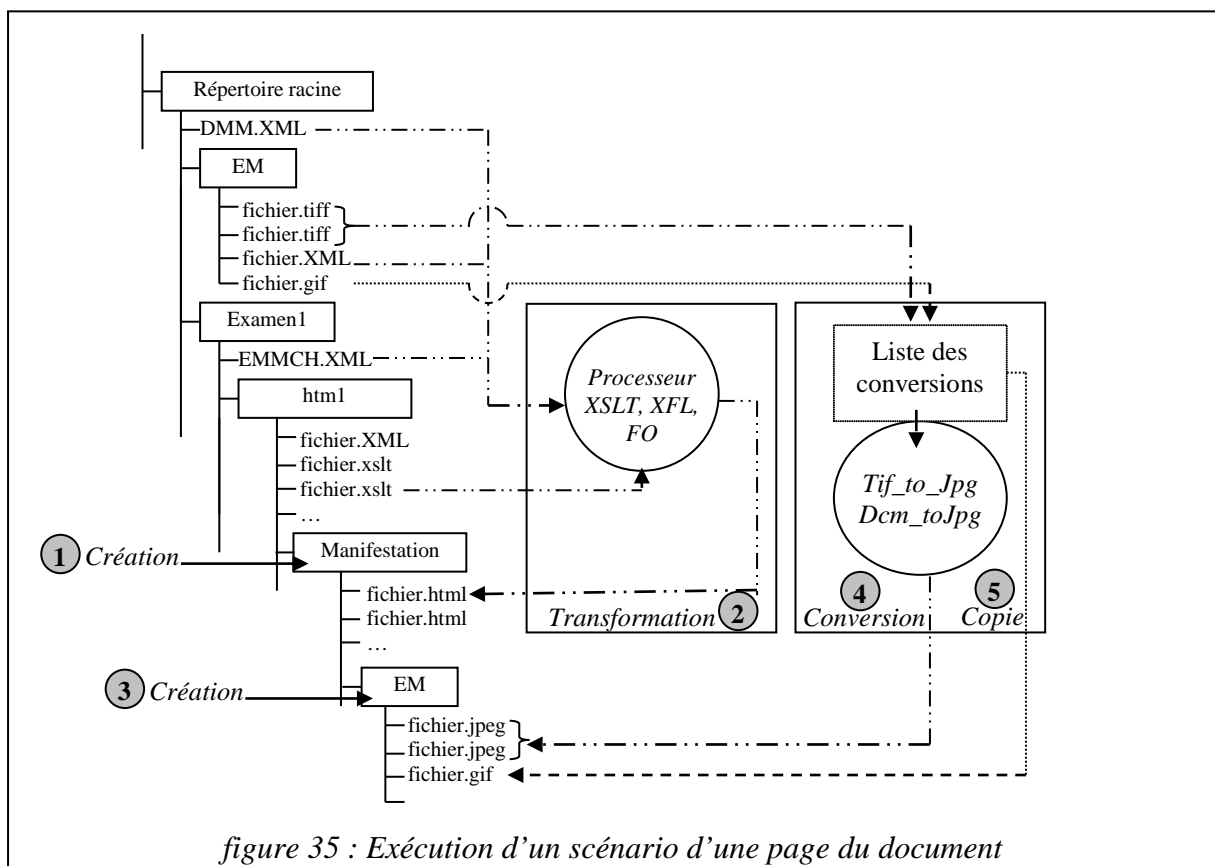


figure 35 : Exécution d'un scénario d'une page du document

La transmission d'un document multimédia médical (VOIR annexe 17, p. 130)

Pour pouvoir transmettre une manifestation d'une page d'un document, la classe « DMMViewer.java » a été simplement étendue afin d'obtenir la classe « DmmSender.java » qui permet cette nouvelle fonctionnalité, via l'introduction d'un nouveau menu « Communication » dans la barre des menus.

La classe « DmmSender.java » utilise la classe « ConfigFrame.java » qui permet d'afficher une interface graphique, destinée à la configuration des paramètres de transmission tels que : les adresses IP, les noms d'utilisateur, les mots de passe, etc. Une fois la configuration établie, elle est enregistrée dans le fichier « Config.ini » qui est placé dans le même répertoire que le fichier « EditerForAnalysis.bat ».

La classe « DmmSender.java » utilise également la classe « SendFrame.java » qui permet d'afficher une seconde interface graphique. Cette interface permet à l'utilisateur d'effectuer, comme il l'a été précisé précédemment, la transmission d'une manifestation d'une page du document soit vers une boîte aux lettres électronique du système « Télémis », soit vers une liste de boîtes aux lettres électroniques conventionnelles (e-mail) via un serveur smtp, soit à une liste destinataires via l'ADICAP. Mais cette interface permet aussi de mettre à disposition une manifestation d'une page du document sur un serveur HTTP fonctionnant en intranet.

Pour réaliser la transmission vers une boîte aux lettres électronique du système « Télémis », le système commence par sélectionner le scénario adéquat, c'est-à-dire celui dont l'attribut « Acces » détient la valeur « htm3 ». Ensuite, il effectue les transformations et les conversions détaillées dans ce scénario. Enfin, il lui suffit d'envoyer via le protocole DICOM, les différentes images sous format DICOM vers le serveur d'acquisition du système « Télémis ». Evidemment, pour pouvoir consulter les fichiers HTML créés lors de la transformation, on insère ceux-ci dans chaque image sous format DICOM, en plaçant leur contenu dans deux « Data Element » dont les « Tag » sont respectivement : « 2523 » et « 2524 ». Le premier « Data Element » renferme le contenu du fichier « DMMExamen1light.htm » et le second « Data Element » renferme le contenu des fichiers « *NomImage.htm* ».

Lors de la transmission vers une liste de boîtes aux lettres électroniques conventionnelles (e-mail), l'utilisateur a l'occasion de définir l'adresse du destinataire, une liste d'adresses de destinataires, le sujet du message et son contenu.

Pour réaliser la transmission proprement-dite, le système sélectionne le scénario adéquat, à savoir celui dont l'attribut « Acces » possède la valeur « htm2 ». Ensuite, il effectue les transformations et les conversions détaillées dans ce scénario. Il existe dès lors un répertoire « Manifestation » contenant l'ensemble des fichiers correspondant à la manifestation produite. Pour réduire la taille mémoire de cette manifestation, une compression « zip » du répertoire est effectuée afin d'obtenir un fichier « Manifestation.zip » compressé. Par la suite, le système joint le fichier compressé au message dont les caractéristiques (adresse du destinataire, adresse des destinataires, sujet et contenu) ont été définies par l'utilisateur. Enfin, le message est transmis via le protocole SMTP au serveur SMTP de l'hôpital pour qu'il dispatche les messages via Internet.

Lors de la transmission vers l'ADICAP, l'utilisateur peut définir la liste des destinataires auxquels il veut demander une expertise. Les destinataires doivent évidemment être abonnés au service proposé par l'ADICAP.

Une fois que cette liste est précisée, l'utilisateur peut réaliser la transmission. Le système sélectionne le scénario adéquat, à savoir celui dont l'attribut « Acces » a la valeur « hmt1 ». Ensuite, il effectue les transformations et les conversions détaillées dans ce scénario. Il existe

dès lors un répertoire « Manifestation » contenant l'ensemble des fichiers correspondant à la manifestation produite dont un fichier « *Numero.htm* » et un fichier « *Numero.ini* » (où *Numero* est le numéro d'enregistrement de l'examen). Il ne manque donc plus qu'un fichier « *Numero.eml* » qui est composé à partir de la liste des destinataires spécifiée par l'utilisateur, pour se conformer aux besoins de l'ADICAP. Le système renomme le répertoire « Manifestation » en « *Numero* » puis compresse le répertoire renommé afin d'obtenir un fichier « *Numero.zip* » obéissant à la spécification de l'ADICAP. Il peut alors transmettre ce fichier au serveur de l'ADICAP, à l'aide du protocole FTP. Cependant, la présence du firewall sur le réseau de l'hôpital, a imposé l'utilisation d'un serveur délégué (proxy) pour effectuer cette dernière opération. On a donc développé, grâce à RMI, un serveur délégué pour le protocole FTP.

Comme on l'a déjà laissé entendre, un quatrième type de transmission a été ajouté aux trois types précédents. En effet, il est également possible de mettre à disposition une manifestation d'une page du document sur un serveur HTTP fonctionnant en intranet.

Un serveur de mise à disposition a donc été développé (VOIR annexe 18, p. 132), reprenant les principes de l'ADICAP. Lors de sa mise en fonction, ce serveur est configuré grâce à un fichier « *ServeurConfig.ini* », afin de déterminer un ensemble de répertoires dans lesquels des documents (fichier « *zip* » respectant la spécification ADICAP, à l'exception de la structure du fichier « *ini* » qui diffère) peuvent être déposés. Chaque ligne du fichier de configuration représente le compte d'un utilisateur sur la machine hébergeant le serveur et comprend le nom d'utilisateur, le mot de passe, l'adresse mail et le répertoire racine de l'utilisateur. Dans sa phase de configuration, le serveur vérifie également l'existence d'un répertoire « *public_html* » dans le répertoire de chaque utilisateur et l'existence d'un fichier « *index.html* » dans le répertoire « *public_html* ». Si le répertoire « *public_html* » et le fichier « *index.html* » n'existent pas, il sont créés. Le fichier « *index.html* » constitue une page d'accueil qui présente la liste des documents qui ont été mis à disposition et fournit pour chaque document les informations suivantes : le nom du document constituant l'hyperlien vers le document lui-même, le nom du patient, le prénom du patient, la date de l'examen, le nom de l'expéditeur, l'adresse mail de l'expéditeur et un lien vers le fichier de type « *zip* » qui contient le document.

Une fois configuré, le serveur inspecte, de manière continue, l'ensemble des répertoires dans lesquels un document sous la forme d'un fichier « *zip* » est susceptible d'être déposé. Lorsque le serveur rencontre un tel fichier, il le déplace vers le sous-répertoire « *public_html* » du répertoire dans lequel le fichier se trouvait et démarre un nouveau processus qui gère la mise à disposition du nouveau document. D'abord, ce processus décompresse le fichier « *zip* ». Ensuite, il vérifie la conformité du fichier « *zip* », c'est-à-dire l'existence d'un fichier « *htm* », d'un fichier « *eml* » et d'un fichier « *ini* ». La structure du fichier « *ini* » est cependant différente de celle décrite par l'ADICAP. Le fichier « *ini* » est simplement composé de trois lignes définissant respectivement le nom du patient, le prénom du patient et la date de l'examen. Si la décompression ou la vérification a échoué, alors un mail d'alerte est envoyé à un administrateur et le document est supprimé. Par contre si ces deux opérations se sont déroulées correctement, le processus met à jour la page d'accueil (fichier « *index.html* ») en ajoutant une ligne dans la liste des documents et envoie un mail à chaque adresse contenue dans le fichier « *eml* » du document. Ce mail a pour seul contenu l'URL du nouveau document, ce qui permet au destinataire d'accéder facilement à celui-ci en employant un client HTTP.

En effet, pour pouvoir visualiser, à distance, les pages d'accueil ainsi que les différents documents mis à disposition (au format HTML), un serveur HTTP (serveur apache) a été installé sur la même machine que le serveur de mise à disposition. Ce serveur HTTP est

configuré de manière à permettre l'accès à chaque compte utilisateur depuis un client HTTP résidant dans l'intranet.

En conclusion, la transmission vers le serveur de mise à disposition s'effectue de la même manière que celle vers l'ADICAP. Cependant, les destinataires ne doivent être abonnés à aucun service. Ils doivent juste posséder une adresse mail valide et, s'ils veulent visualiser le document, un client HTTP fonctionnant en intranet. Une seconde différence par rapport à la transmission vers l'ADICAP est que le destinataire doit posséder un compte sur la machine hébergeant le serveur et doit figurer dans la liste du fichier de configuration du serveur de mise à disposition. La dernière différence est que le transfert du fichier « zip » à l'aide du protocole FTP ne doit plus être réalisé via un serveur délégué (proxy) puisque le serveur de mise à disposition est situé dans l'intranet.

L'édition d'un document multimédia médical (VOIR annexe 17, p. 131)

En ce qui concerne l'édition d'un document multimédia médical, la décision a été prise de se limiter à la simple édition des informations médicales contenues dans les fichiers XML relatives au document et à la page (*figure 29*). L'outil d'édition se restreint donc à ces opérations rudimentaires.

Pour pouvoir éditer un document, la classe « DMMSender.java » a simplement été étendue afin d'obtenir la classe « DmmEditerForAnalisys.java » qui permet cette nouvelle fonctionnalité, via l'introduction d'un nouveau menu « Edition » dans la barre des menus.

Comme on l'a exprimé précédemment, il est intéressant d'avoir plusieurs vues du document lors de l'édition. Quatre vues ont été identifiées : la vue de présentation, la vue du scénario, la vue du storyboard et la vue des objets. La vue présentation est présente puisque l'utilisateur peut déjà visualiser les différentes manifestations au format HTML. La vue du scénario et la vue du storyboard ne sont pas nécessaires puisque la dimension temporelle est absente de nos manifestations. Par contre, il est avantageux de proposer la vue des objets contenus dans le document. C'est pourquoi la classe « DmmEditerForAnalisys.java » utilise la classe « EditerGalleryForAnalisys.java » afin d'offrir un aperçu des images contenues dans le document, sous forme d'une galerie d'images et d'obtenir les observations faites sur ces mêmes images.

Mais la classe « EditerGalleryForAnalisys.java » permet également de gérer l'édition des informations et d'afficher les interfaces graphiques, destinées à l'introduction des nouvelles informations.

Cette classe fonctionne de telle manière que deux versions du document actuellement en cours de modification sont toujours conservées : une version courante (composée d'une instance de la classe « DMM.java », nommée « CurrentDmm » et d'une instance de la classe « EMMCH.java », nommée « CurrentEmmch »), et la dernière version ayant été sauvegardée (composée d'une instance de la classe « DMM.java », nommée « LastSavedDmm » et d'une instance de la classe « EMMCH.java », nommée « LastSavedEmmch »). A l'ouverture du document (VOIR La visualisation d'un document multimédia médical, p. 58), ces deux versions sont identiques. Ensuite, toutes les modifications d'informations ne sont répercutées que sur la version courante. Les deux versions ne seront à nouveau identiques que si l'utilisateur décide de sauver les modifications effectuées. Cette sauvegarde peut être accomplie soit à tout moment en pressant l'attribut adéquat dans le menu, soit lors de l'ouverture d'un nouveau document ou lors de la fermeture du document actuellement en cours d'édition. Dans ces deux dernières situations, une boîte de dialogue est affichée afin que l'utilisateur puisse choisir s'il désire sauvegarder les modifications ou non.

Les informations contenues dans les fichiers « XML » du document sont, quant à elles, toujours identiques à celles de la version courante. En effet, puisqu'il est supposé que le

paradigme « WYSIWYG » doit être mis en œuvre, l'utilisateur doit, pour chaque opération d'édition effectuée, avoir l'opportunité de percevoir immédiatement son effet sur la présentation. Donc, lors de chaque modification des informations, la manifestation HTML présentée doit être régénérée et réaffichée. Puisque cette manifestation est générée à partir des fichiers « XML » du document, ceux-ci doivent évidemment être à jour.

En conséquence, lors de l'ouverture d'un nouveau document ou lors de la fermeture du document en cours d'édition, si l'utilisateur décide de ne pas sauvegarder les modifications, les fichiers « XML » sont réécrits afin de contenir les informations identiques à celles de la dernière version sauvegardée.

En conclusion, le processus d'édition fonctionne selon un cycle d'action puis de perception, où à chaque étape, l'utilisateur réalise une modification sur le document puis le système applique cette modification et présente à l'utilisateur le document modifié. En effet, les informations à éditer sont d'abord introduites grâce à une interface graphique, puis l'utilisateur confirme leurs modifications. Suite à cette confirmation, les informations qui sont contenues dans l'instance des classes « DMM.java » et « EMMCH.java », sont effectivement modifiées. Pour les raisons évoquées précédemment, les informations correspondantes inscrites dans les fichiers « XML » sont également modifiées. Pour finir, le système ré-exécute le scénario courant afin d'obtenir une nouvelle manifestation qui est affichée, prenant en compte les modifications.

L'analyse automatique de lames

Le troisième bouton ajouté à l'interface d' « Analysis » (VOIR annexe 13, p. 115), permet d'entamer l'analyse automatique de 1 à 8 lames se trouvant sur la platine du microscope.

Grâce à ce bouton, l'utilisateur obtient une boîte de dialogue permettant l'identification des lames à analyser. Cette identification doit être effectuée grâce au numéro associé au code barre se trouvant sur la lame. L'utilisateur introduit donc pour chaque lame, le numéro d'identification dans la case correspondant à la place de la lame sur la platine du microscope, (à l'aide du clavier). L'utilisateur peut ensuite confirmer la poursuite de la procédure d'analyse. Suite à cette confirmation, le système vérifie, grâce au module « Gestion Analyse/Patient », s'il existe, pour chaque lame, une demande d'analyse dont le numéro unique est identique au numéro introduit. La demande d'analyse (*figure 36*) étant un fichier codé en caractères ASCII d'une seule ligne et nommé par son numéro unique suivi de l'extension « .daa » (pour demande d'analyse automatique), il suffit de rechercher, dans un répertoire déterminé, un fichier nommé par le numéro introduit suivi de l'extension « .daa ». La ligne contenue dans un fichier de demande contient le numéro unique, le nom du patient, le prénom du patient, sa date de naissance et la date de prélèvement. Chacune de ces informations ne peut contenir aucun caractère « espace » (code ASCII 032) et est séparée de l'information suivante par le caractère « ; » (code ASCII 059).

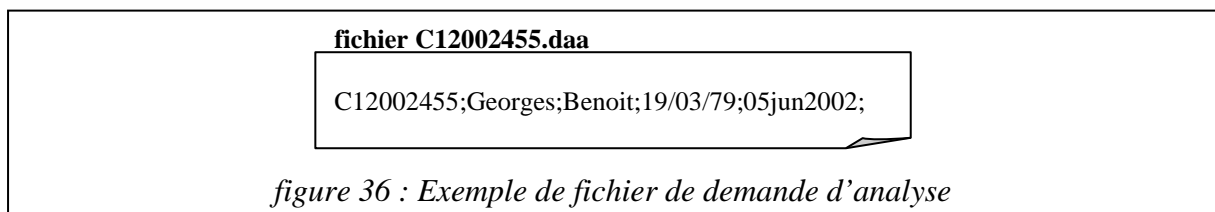


figure 36 : Exemple de fichier de demande d'analyse

Il faut également remarquer que lors de la prise de connaissance des demandes d'analyse, le système central de gestion des dossiers fournit un fichier codé en caractères ASCII et énumérant les demandes sous forme d'une suite de lignes, chacune indiquant le numéro unique, le nom du patient, le prénom du patient, sa date de naissance et la date de

prélèvement. Dès lors, il faut, pour chaque ligne de ce fichier, créer un fichier de demande d'analyse automatique contenant la ligne traitée en vérifiant l'absence de caractères « espace » et la présence des caractères « ; ».

En conséquence de la vérification, s'il existe un numéro introduit auquel aucun fichier de demande ne correspond, un message d'erreur signalant l'absence de cette demande est affiché et l'utilisateur à l'opportunité d'introduire un nouveau numéro. Sinon, les informations contenues dans les fichiers sont récupérées et associées aux numéros et le processus d'analyse peut continuer.

Le système demande alors à l'utilisateur de calibrer chaque lame. La calibration consiste à déterminer une position de la platine en X et en Y, à partir de laquelle l'observation de lame débute. Une fois l'ensemble des calibrations effectuées, une boîte de dialogue est affichée afin que l'utilisateur puisse confirmer définitivement le déclenchement de l'examen des lames. Si l'examen est déclenché, le système va traiter les lames une à une, grâce au module « Automatisation Analyse des lames », avant d'afficher un message de fin d'analyse.

Pour chaque lame, le système communique avec le microscope afin de placer l'objectif 20X, grâce au pivot motorisé. Ensuite, toujours via le canal de communication, il positionne la platine motorisée, suivant les coordonnées enregistrées lors de la calibration. Une fois la platine positionnée, le système acquiert le premier champ d'observation, c'est-à-dire l'image qui peut être observée à partir de ce positionnement. Le système détecte, suivant la valeur donnée aux paramètres de la configuration générale de détection, des groupes de particules et répertorie leur position dans une grille dont chaque ligne représente une position. Cependant, ces positions sont recalculées afin de centrer la zone d'intérêt. Les nouvelles positions sont alors définitivement enregistrées dans une nouvelle grille « Stage Position ». Après cet enregistrement, le système déplace à nouveau la platine afin d'obtenir le deuxième champ d'observation contigu au premier. Le système réalise les mêmes opérations pour ce deuxième champ d'observation que pour le premier et enregistre les positions recalculées à la suite de celles déjà inscrites. Le système continue de la même manière pour tous les champs d'observation de la zone à observer. Or, la zone à observer est déterminée par la position de départ de l'observation et le nombre de champs en X et en Y à observer.

Lorsque le système a balayé l'ensemble de la zone d'observation, il dispose d'une grille mentionnant un ensemble de positions. Il place alors l'objectif 100X et déplace la platine du microscope aux différentes positions répertoriées afin d'acquérir, dans les buffers d'Analysys les images englobant les régions d'intérêt. Les images acquises sont ensuite découpées afin de réduire leur taille mémoire. Après ce découpage, les images sont stockées dans des fichiers au format TIFF non compressés et placées dans un sous-répertoire, nommé « EM » d'un répertoire, nommé par le numéro unique de la lame que le système est actuellement en train de traiter (ces répertoires étant bien sûr créés précédemment).

Ensuite, le système entreprend la composition d'un document multimédia médical, contenant un seul examen regroupant les informations de la demande d'analyse (le numéro unique, le nom du patient, le prénom du patient, sa date de naissance et la date de prélèvement) et les images. Si l'utilisateur en a exprimé le souhait, le système envoie le document composé vers le serveur « Télémis ».

Etant donné que les modules « Automatisation Analyse des lames » et « Composition des documents » ou « Transmission des documents » ont été réalisés dans deux langages différents, il a fallu utiliser la même astuce que pour les cas précédents. Le système doit lancer, à partir du module « Automatisation Analyse des lames », un "shell" auquel un fichier « LauncherForAnalysis.bat » est fourni et qu'il doit interpréter. Des arguments, en l'occurrence le numéro unique, le nom du patient, le prénom du patient, sa date de naissance, la date de prélèvement, le répertoire dans lequel se trouvent les images et un indicateur

déterminant si oui ou non l'envoi automatique vers le serveur « Télémis » doit être effectué sont également associés au "shell". Le fichier « LauncherForAnalysis.bat » permet simplement d'activer la machine virtuelle Java sur la classe « Launcher.java » qui utilise évidemment les arguments afin de composer et transmettre si nécessaire le document.

La paramétrisation de l'analyse automatique de lames

Le quatrième bouton ajouté à l'interface d' « Analysis » (VOIR annexe 13, p. 115), permet d'accéder à une boîte de dialogue autorisant le choix des paramètres suivants : le chemin du répertoire dans lequel les nouveaux documents composés sont placés, le chemin du répertoire dans lequel les demandes d'analyses sont placées, le chemin du répertoire dans lequel les demandes d'analyses sont transférées une fois effectuées et enfin, les paramètres concernant l'automatisation (le nombre de champs en X et en Y observés, le nombre maximum de cellules sélectionnées et le choix de savoir si oui ou non l'envoi automatique vers « Télémis » est effectué). La valeur attribuée à ces paramètres est conservée dans le fichier « analysis.ini » spécifique au logiciel « Analysis ». Si aucune valeur n'est disponible pour l'un de ces paramètres, une valeur par défaut lui est attribuée.

6.2.2. Le « Télémis reception »

Grâce au module « Interface Télémis », un menu (VOIR annexe 14, p. 116) a été ajouté à l'interface affichant les images sous forme de galerie, du module « Télémis-Reception ». Ce menu autorise les fonctionnalités suivantes : la composition d'un document, sa visualisation, sa transmission et son édition.

Ces fonctionnalités sont réalisées de la même façon que pour leur mise en œuvre dans « Analysis », à quelques différences près.

Premièrement, les modules « Interface Télémis » et « Composition des documents », « Transmission des documents » ou « Edition des documents » ont été développés dans le même langage (java), il est donc possible d'utiliser les composants entre eux.

Deuxièmement, lors de la composition d'un nouveau document, les images associées au document sont celles qui sont sélectionnées dans l'interface du module « Télémis-Reception ». En effet, il est possible de sélectionner un certain nombre d'images dans la galerie affichée.

Quatrième partie : Discussion

7. Les lacunes du système

La première faiblesse du système est l'omission du rôle que joue la sécurité dans un tel projet. En effet, les documents composés sur un disque local de l'ordinateur de la station d'acquisition, sont accessibles à toute personne pouvant utiliser cet ordinateur.

La transmission d'une manifestation vers le serveur HTTP fonctionnant dans l'Intranet n'est pas sécurisée. Cela n'est pas trop inquiétant, étant donné la présence du firewall qui isole le réseau interne. Cependant le serveur HTTP permet à toute personne possédant un client HTTP (fonctionnant dans l'Intranet) et connaissant l'URL à laquelle sont disponibles les manifestations, de visualiser celles-ci sans devoir s'identifier préalablement. De plus, les transmissions vers l'ADICAP ou via e-mail empruntent l'Internet sans aucun dispositif de sécurité particulier.

Il faut également remarquer que l'on a considéré le domaine d'application de l'informatique dans la problématique de la télémicroscopie en cytologie hématologique à l'UCL Mont-Godinne comme étant isolé du gestionnaire des dossiers des patients. Or, lors de l'analyse automatique d'une lame, le système vérifie l'existence d'une demande d'analyse correspondante et représentée par un fichier ASCII. Il serait donc utile d'automatiser également l'envoi de ces demandes à partir du gestionnaire des dossiers vers la station d'acquisition afin d'éviter au laborantin de devoir effectuer une requête au gestionnaire de dossiers, de récupérer le fichier résultant de la requête, de formater ce fichier en plusieurs fichiers de demandes d'analyse (un pour chaque ligne) et de transférer ces demandes dans un répertoire de la station d'acquisition.

Si l'on considère la composition et l'édition d'un document, on observe que les utilisateurs ne peuvent exécuter que des opérations limitées. Ils peuvent uniquement changer la valeur que prennent les informations du document et de la page. Lors de la composition, ils doivent se limiter à quelques scénarios imposés par défaut. L'édition des scénarios déjà présents pour une page est impossible. L'ajout d'une nouvelle page au document ou d'un nouveau scénario à une page est également irréalisable. Aucune manipulation ne peut être effectuée sur les médias contenus dans le document. Bref, les manipulations autorisées sur les documents sont insuffisantes. Mais ceci peut s'expliquer par le fait que l'on s'est limité à la problématique de la télémicroscopie pour laquelle une galerie d'images était suffisante.

Les différentes dimensions d'un scénario, logique, spatiale, temporelle et hypermédia, qui auraient dû faire chacune l'objet d'une feuille de transformation, ont été fusionnées. De ce fait, toute édition d'une dimension particulière est impossible. La création de DTD, permettant de définir ces dimensions a été omise. Ainsi des difficultés de composition et d'exécution concernant ces scénarios ont pu être évitées. Cependant, la manière de constituer les scénarios utilisée a permis de comprendre le principe de fonctionnement de ceux-ci.

L'utilisation du système « Télémis » est bénéfique car ce système allie sécurité, performance de transfert, facilité et efficacité de visualisation des images. Cependant, le système ne permet aucune édition et les documents qui y sont échangés sont de simples regroupements d'images. Il serait intéressant de créer un système semblable permettant, d'une part, l'échange de documents plus évolués, tels que ceux conçus dans ce projet, et, d'autre part, l'édition de ces mêmes documents.

Enfin, on peut également remarquer que la visualisation des images en provenance du microscope est réalisée, grâce à « Télémis », au travers d'une galerie d'images. Cependant on

peut regretter que cette manière de visualiser ne soit pas plus proche de la façon dont on visualise les images aux travers des oculaires d'un microscope. On pourrait imaginer un visualisateur dédié à la microscopie, dont chaque fonctionnalité serait une métaphore de la réalité. Que ce soit un système statique ou dynamique, il suffirait de permettre la visualisation comme une image représentant le champ d'observation tel qu'il pourrait être vu au travers des oculaires. On pourrait changer d'objectif, se déplacer de bas en haut, de gauche à droite, tout comme si on observait une lame à partir du microscope lui-même.

Il faut remarquer que dans le cas d'un système dynamique, la mise en œuvre d'une telle visualisation est plus simple, puisqu'il suffirait de capturer en temps réel le champ d'observation et de répercuter les manœuvres effectuées à distance sur le microscope. Tandis que pour un système statique, étant donné que les images doivent être stockées puis visualisées à posteriori, il faudrait stocker l'ensemble de la surface de la lame, à chaque grossissement, sous forme d'images. Cela correspondrait à un espace mémoire gigantesque.

8. Le système « idéal »

On pourrait donc développer un meilleur système que celui conçu précédemment, qui serait constitué de quatre composants (*figure 37*).

Le premier composant, nommé « Serveur d'Acquisition », permettrait de composer un document contenant une page regroupant un ensemble d'éléments multimédia et d'informations médicales qui lui sont transmis et contenant un seul scénario pré-défini, valable pour n'importe quel type d'examen. La manifestation résultant de l'exécution de ce scénario permettrait une visualisation, par défaut, de la page du document. Lors de cette composition, un groupe d'accès devrait être défini et inscrit d'une part pour le document et d'autre part pour la page. En effet, puisque le document est destiné à être placé sur un serveur et donc à être potentiellement accessible à de nombreux utilisateurs, le groupe d'accès inscrit dans le document permet de définir les utilisateurs qui y auront effectivement accès. Une fois le document composé, le « Serveur d'Acquisition » devrait le transmettre au deuxième composant. Dans ce cas, ce n'est pas une manifestation de la page qui est envoyée, mais bien le document lui-même.

Le second composant, appelé « Serveur de Document », devrait être capable de stocker les documents qui lui sont expédiés, mais il devrait également permettre aux utilisateurs ayant accès aux documents de les éditer via le troisième composant, le « Client d'Édition ». Bien entendu, l'édition d'un document implique sa visualisation. Pour éditer un document, l'utilisateur devrait d'abord s'authentifier afin d'avoir accès à un certain nombre de documents et plus précisément un certain nombre de pages. Lorsque l'utilisateur aurait émis le souhait d'éditer un document particulier, ce document serait transféré vers le « Client d'Édition » mais avec uniquement les pages auxquelles il a accès. Puisque plusieurs utilisateurs ont accès aux mêmes pages, il y a une certaine concurrence entre ceux-ci. Le serveur, pour gérer cette concurrence, indiquerait pour le document et pour chaque page son statut, c'est-à-dire si le document ou la page est ouvert en lecture, en écriture ou fermé. Une fois le document transféré, l'utilisateur aurait l'opportunité de réaliser un ensemble de modifications sur ce dernier. Ces modifications devraient évidemment être répercutées sur le document stocké sur le serveur. Le « Serveur de Document » devrait également être à même de fournir les diverses manifestations des pages de documents à un quatrième composant, le « Visualisateur de Document », afin de permettre la visualisation des manifestations. Dans ce cas, l'utilisateur ne doit pouvoir visualiser que les manifestations qui lui sont réservées et adéquates à l'environnement dans lequel il évolue et à son profil. Le choix des scénarios

convenant pour une situation particulière pourrait être fait sur l'attribut « Acces ». Si celui-ci diffère pour chaque scénario, il suffit d'associer l'accès avec un certain nombre de conditions à respecter. Ces conditions devraient, probablement être définies lors de la composition des scénarios. Bien entendu, l'ensemble des communications entre les différents composants nécessite la conception de nouveaux protocoles permettant d'assurer l'échange d'informations telles que l'environnement dans lequel l'utilisateur évolue, son profil, etc.

On pourrait également essayer de développer ce serveur de telle manière qu'il supporte le protocole HTTP et qu'il puisse fournir, via ce protocole, des manifestations au format HTML à un client HTTP conventionnel tel que « Netscape » ou « Internet Explorer ».

Le troisième composant « Client d'Édition » devrait permettre les fonctionnalités suivantes :

- Composition d'un document sans page
- Ajouter une page (contenant un scénario par défaut) à un document
- Supprimer une page à un document
- Ajouter un élément multimédia à une page
- Supprimer un élément multimédia à une page
- Ajouter un scénario prédéfini à une page
- Ajouter un nouveau scénario à une page
- Supprimer un scénario à une page
- Ajouter un élément multimédia à un scénario
- Supprimer un élément multimédia à un scénario
- Définir le positionnement d'un élément multimédia dans un scénario
- Définir les caractéristiques temporelles d'un élément multimédia dans un scénario
- Définir la dimension hypermédia dans un scénario
- Manipuler un élément multimédia (par exemple, si c'est une image : zoomer, placer des formes, changer les couleurs, etc.)

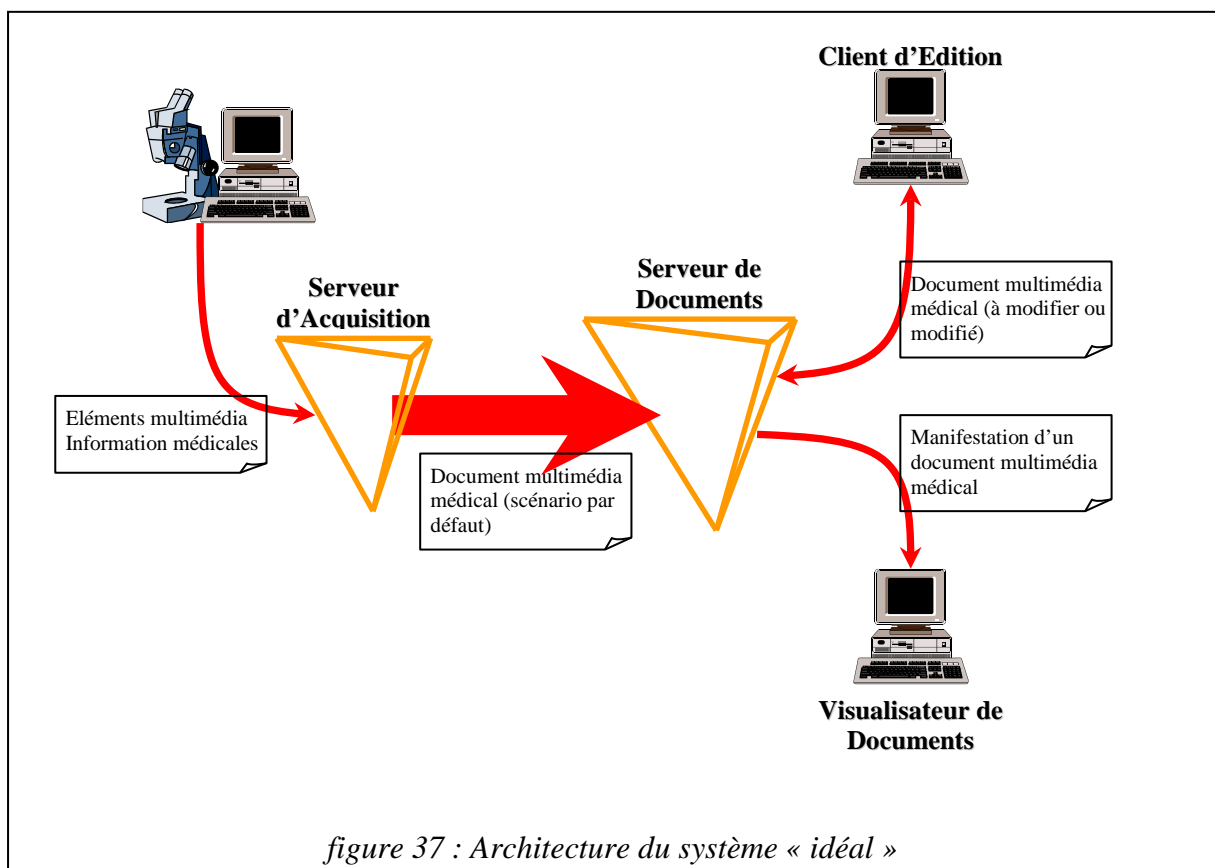
Pour mettre en œuvre l'édition d'un document multimédia, telle que décrite à la fin du point 2.5.4, il faudrait sans doute fournir séparément un moyen d'édition du contenu du document et un moyen d'édition des scénarios associés aux pages des documents.

Le moyen d'édition du contenu du document permettrait dès lors de modifier les informations contenues dans les fichiers XML initiaux spécifiant la dimension logique du document. Ce moyen devrait donc permettre les fonctionnalités de composition d'un document sans page, d'ajout d'une page (contenant un scénario par défaut) à un document, de suppression d'une page à un document, d'ajout d'un élément multimédia à une page, de suppression d'un élément multimédia à une page, d'ajout d'un scénario prédéfini à une page, de suppression d'un scénario à une page.

Le moyen d'édition des scénarios permettrait de créer un nouveau scénario ou de modifier un scénario déjà existant. Ce procédé devrait donc aussi permettre d'ajouter un élément multimédia à un scénario, de supprimer un élément multimédia à un scénario, de définir le positionnement d'un élément multimédia dans un scénario, de définir les caractéristiques temporelles d'un élément multimédia dans un scénario, de définir la dimension hypermédia dans un scénario, de manipuler un élément multimédia (par exemple, si c'est une image : zoomer, placer des formes, changer les couleurs, etc.). L'architecture d'un tel système d'édition de scénarios pourrait être axée autour des quatre vues identifiées au point 2.5.6 afin de faciliter la modification des dimensions : spatiale, temporelle et hypermédia.

Pour ajouter un nouveau scénario à une page, il suffirait simplement de créer un nouveau fichier XML comprenant les informations du scénario et créer des feuilles de transformation pour chaque dimension, afin d'associer le tout à une page. Tandis que pour définir ou modifier les différentes dimensions du nouveau scénario ou d'un scénario existant, il s'agirait dès lors de modifier les feuilles de transformation, afin que ces feuilles de transformation correspondent aux modifications des dimensions formulées. Cette correspondance entre modifications des feuilles de transformation et modifications des dimensions semble cependant assez difficile à réaliser.

Le dernier composant, appelé « Visualisateur de Document », devrait permettre de visualiser les manifestations des pages des documents qui sont adéquates à l'environnement dans lequel l'utilisateur évolue et à son profil. Dans le domaine médical, chaque discipline a ses propres besoins en terme de présentation. Donc, il faudrait créer un visualisateur adapté pour chaque discipline médicale. Bien entendu, ces visualisateurs sont destinés à interpréter les manifestations qui leurs sont transmises par le « Serveur de Document ». En conséquence, il faudrait peut-être créer une manifestation dédiée à chaque visualisateur ou chaque discipline. On pourrait également concevoir que les navigateurs tels que « Internet Explorer » ou « Netscape » puisse jouer, par défaut, le rôle de ce dernier composant. Cela pourrait donc engendrer une manifestation HTML pour chaque domaine médical, pouvant ainsi être interprétée par les navigateurs.



Conclusion

Mon séjour au laboratoire de cyto-hématologie à l'UCL Mont-Godinne m'a permis d'apprécier les avantages que les développements informatiques récents peuvent apporter à l'établissement des diagnostics émis à partir d'une analyse cyto-hématologique.

En effet, les praticiens confrontés à des cas cliniques qui dépassent le cadre de leur savoir souhaitent pouvoir consulter des experts situés dans d'autres cliniques du pays ou à l'étranger. Ils souhaitent également pouvoir comparer l'observation microscopique qui leur est demandée à des observations antérieures faites sur le même patient ou sur d'autres patients, dans leur laboratoire ou dans un autre. L'automatisation des analyses cyto-hématologiques fait également partie des attentes énoncées par les praticiens. L'informatisation offre, de toute évidence, des possibilités en ce qui concerne la transmission à distance des données, leur conservation en bon état et leur ré-observation.

Grâce au logiciel « Télémis » et au réseau de communication déjà mis en place à l'UCL Mont-Godinne, les praticiens avaient l'opportunité d'échanger et de consulter les images provenant de plateaux techniques d'acquisition tels que la « Magnetic Resonance », la « Computed Tomography », la « Positron Emission Tomography », etc.

Indépendamment de cette possibilité, le logiciel « Analysis », couplé au microscope « AX70 Olympus », permettait la sélection, l'acquisition et le stockage manuel d'images provenant de ce dernier.

Cependant, les images sélectionnées et acquises ne pouvaient être échangées et consultées à distance. Aucun moyen n'était mis en œuvre pour permettre le transfert de ces images vers le système « Télémis » ou vers un autre système de diffusion et d'échange.

Pour remédier à ces insuffisances, j'ai donc pu développer un système utilisant l'infrastructure et les logiciels disponibles. Ce système permet de composer un document médical multimédia regroupant des images sélectionnées et acquises grâce à « Analysis » et les informations médicales associées à ces images. A partir du document composé, il rend possible la transmission d'une manifestation de celui-ci, via le système « Télémis », via l'« ADICAP » ou via e-mail.

Le système autorise également la réalisation automatique des analyses cyto-hématologiques grâce à un module particulier développé dans le logiciel « Analysis ». Ce module permet de sélectionner et d'acquérir automatiquement un ensemble d'images représentatives de l'échantillon observé. Le processus d'automatisation se poursuit par l'intégration des images et des données médicales associées, sous forme de documents permettant les différents types de transmission.

Bien que le système élaboré réponde aux attentes formulées par les spécialistes, en termes d'échange et de consultation à distance de données, il comporte certaines lacunes. Les améliorations à entrevoir portent sur la sécurité, les manipulations d'édition autorisées sur les documents et la richesse de ceux-ci. En effet, le système « Télémis » ne propose aucun moyen d'édition des documents qui y sont échangés et ces derniers sont de simples regroupements d'images. De plus, les documents conçus ont été simplifiés par la fusion des différentes dimensions d'un scénario, rendant impossible toute édition d'une dimension particulière.

Afin d'éviter ces faiblesses il serait utile de concevoir un système dont les bases ont été brièvement exposées. On peut également rêver de développer un système dynamique, permettant de prendre en main à distance le microscope. Cependant, il faudra vraisemblablement attendre que les contraintes techniques évoluent.

Références

[ADICAP00] CRIHAN, ‘‘Messagerie ADICAP, spécifications techniques et fonctionnelles’’, 2000, <http://www.adicap.asso.fr>.

[AHL96] Nabil R. Adam, Richard D. Holowczak et Wen-Syan Li, ‘‘Object manifestations for medical imaging application’’, *Computerized Medical Imaging and Graphics*, Vol. 20, No. 4, pp301-314, 1996.

[Aldus92] Aldus Corporation, ‘‘TIFF 6.0 Specification’’, 1992, <http://susdl.fcla.edu/strucmeta/tiff.html>.

[Apache99] Apache Software Foundation, ‘‘Technologies employées’’, 1999, <http://www.epita.net/~cococon/technologies.html>.

[AR95] AJ Ahumada, Horng Rensheng, ‘‘Smoothing DCT Compression Artifacts’’, *Society of information display of technical papers*, Vol.25, pp. 708-711, 1995.

[BFWAMS99] Erwin Bellon, Michel Feron, Joost Wauters, Werner Aerts, Guy Marchal et Paul Suetens, ‘‘Advanced software for teleradiology : exploiting Java applets and the www to make multimédia tele-access and tele-interaction universally available’’, *European telemedecine*, pp. 66-71, 1998-1999.

[BJK00] Frederic Bes, Muriel Jourdan et Farid Khantache, ‘‘A generic architecture for automated construction of multimedia presentation’’, *INRIA*, 2001.

[BKW00] Susanne Boll, Wolfgang Klas et Utz Westermann, ‘‘Multimedia Document Models’’, *Multimedia Tools and Applications, ICMCS Special Issue*, 2000.

[CRID00] CRID FUNDP, Guide à destination des utilisateurs d’Internet, 2000, http://mineco.fgov.be/information_society/consumers/consumers_internetguide/Userguide_fr.htm.

[Crihan00] CRIHAN, Spécifications techniques et fonctionnelles de l’ADICAP, 2000, <http://www.adicap.asso.fr/DOC-TECH/specificationsadicap.pdf>.

[DBBCDEF02] Francesca Demichelis, Mattia Barbareschi, Sebastiana Boi, Claudio Clemente, Paolo Dalla Palma, Claudio Eccher, Stefano Forti, ‘‘Robotic Telepathology for Interoperative remote diagnosis using still-imaging-based system’’, 2002.

[DellaMea99] Vincenzo Della Mea, ‘‘Store-and-forward telepathology’’, *European telemedecine*, pp. 74-76, 1998-1999.

[Derhy] Serge Derhy, ‘‘Norme Dicom en imagerie médicale : principe d’utilisation, vérités et idées reçues, utilisation d’un décodeur d’images en langage java’’, 1998, <http://eviewbox.sourceforge.net/JFR98/>.

[Dorban99], Hugues Dorban, ‘‘Etude exploratoire du marché belge de la télépathologie : Le cas de la société Télémis’’, Mémoire en vue de l’obtention du titre de licence, Institut d’Administration et de gestion, UCL, 1999.

[Emim02] Hubert Meurisse, Jean Fichet, ‘‘Aide au diagnostic, Edition multimédia et sécurité’’, *Projet EMIM2*, 2000-2002.

- [Holman00] Ken Holman, “What is XSLT”, 2000,
<http://www.xml.com/pub/a/2000/08/holman/index.html>.
- [HPBPC] Steven C. Horiil, Fred W. Prior, W. Dean Bidgood, Charles Parisot, Geert Claeys,
“DICOM : An Introduction to the Standard”, <http://www.dicomanalyser.co.uk/html/introduction.htm>.
- [Jacobsson99] Ulf Jacobsson, “Telemedicine collaboration in the Baltic Sea region”,
European telemedicine, pp. 129-132, 1998-1999.
- [JRT98] M. Jourdan, C. Roisin, L. Tardif, “Multiviews Interfaces for Multimedia Authoring
Environnements”, *Proceedings of the 5th Conference on Multimedia Modeling*, pp. 72-79,
1998.
- [Kaddour99] Kaddour Chakib, “Compression des images fixes par fractales basée sur la
triangulation de Delannay et la quantification vectorielle”, 1999,
<http://iquebec.iframe.com/kadchakib>.
- [Layaida97] Nabil Layaida, “Madeus : système d’édition et de présentation de documents
structurés multimédia”, *Thèse, Université Joseph Fourier Grenoble*, 1999.
- [Leong01a] FJ Leong, “Telepathology terminology explained”, 2001,
<http://www.ndp.ox.ac.uk/telepathology/definitions.htm>.
- [Leong01b] FJ Leong, “Classification system”, 2001,
<http://www.ndp.ox.ac.uk/telepathology/classifications.htm>.
- [LGGM99] FJ Leong, Andrew Graham, Thomas Gahm, James O’D McGee,
“Telepathology : Clinical utility and methodology”, *Recent Advances in Histopathology 18*,
1999, http://www.ndcls.ox.ac.uk/telepathology/review_1999.htm.
- [LM01] FJ Leong, James O’D McGee, “Automated complete slide digitization : a medium
for simultaneous viewing by multiple pathologists”, *Journal of Pathology*, pp. 508-514,
2001.
- [Mercier00] Benoît Mercier, “Mise au point d’une station de télé-médecine pour le laboratoire
d’hématologie”, *Mémoire en biologie médicale, Institut Paul Lambin, UCL*, 1999-2000.
- [Nema99] Nema, “Digital Imaging and Communications in Medicine (DICOM) Part 1-14”.
- [Nordrum99] Ivar Nordrum, “History and present status of real-time telepathology”,
European telemedicine, pp. 72-73, 1998-1999.
- [OPERA00] Les membres du projet OPERA, “Outils pour les documents électroniques,
recherche et applications”, *Rapport d’activité, Projet Opéra, INRIA*, 2000.
- [Rensburg98] Ado van Rensburg, “Making TIFF DICOM compatible”, 1998,
<http://www.up.ac.za/saapmb38/ado/>.

[Roisin99] Cécile Roisin, “Documents multimédia structurés”, *Habilitation à diriger des recherches, spécialité informatique, Institut National Polytechnique*, 1999.

[Stanberry99] Ben Stanberry, “Legal and ethical issues in European telemedecine”, *European telemedecine*, pp. 20-25, 1998-1999.

[Tanenbaum99] Andrew Tananbaum, Réseaux Cours et exercices, 3^{ième} Edition, 1999.

[Telemis00] Telemis, “Telemis Medical”, Telemis Reception User Manual 3.3, 2000.

[TSDLB99] Sapal Tachakra, Alganandan Sivakumar, Jesus Diaz-Guijarro Hayes, Maria Lynch et Julie Bak, “The use of telemedecine for remote trauma management in minor accident and treatment services”, *European telemedecine*, pp. 103-104, 1998-1999.

[VRL00] Lionel Villard, Cécile Roisin and Nabil Layaida, “An XML-based multimedia document processing model for content adaptation”, *Digital Documents and Electronic Publishing, LNCS, Projet Opéra, INRIA*, 2000

[W3C98] W3C Recommendation, “Extensible Markup Language (XML)1.0”, 1998, <http://www.w3.org/TR/1998/REC-xml-19980210>.

[W3C99a] W3C Working Draft, “Extensible Stylesheet Language (XSL) Specification”, 1999, <http://www.w3.org/TR/WD-xsl>.

[W3C99b] W3C Recommendation, “Extensible Stylesheet Language Transformation (XSLT) 1.0 Specification”, 1999, <http://www.w3.org/TR/1999/REC-xslt-19991116>.

[W3C01] W3C Recommendation, “XML Linking Language (XLink) 1.0 Specification”, 2001, <http://www.w3.org/TR/xlink/>.

[Watson95] Andrew B Watson, “A technique for visual optimization of DCT quantization matrices for individual images”, *Society of information display of technical papers*, Vol.14, pp. 946-949, 1995.

Annexes

Annexe 1 : Les méthodes de compressions sans perte

La méthode de compression sans perte la plus connue est celle de « Lempel Ziv et Welch » (LZW). Cette méthode est supportée par les formats TIFF et GIF. Elle utilise une table qu'elle construit dynamiquement, au cours de la compression et de la décompression, qui n'est pas stockée dans le fichier compressé. Lors de la compression, elle initialise la table avec 258 entrées dont les 256 premières (0-255) sont réservées pour les caractères ASCII. Ensuite elle lit les données numériques comme des caractères ASCII. Elle trouve la plus longue entrée correspondant avec les n premiers caractères du début des données. Elle enregistre l'index correspondant dans le fichier compressé et ajoute une entrée dont la valeur comprend les n+1 premiers caractères du début des données. Enfin elle omet les n premiers caractères, pour ne prendre en compte que les caractères non traités. Lors de la décompression, on effectue le raisonnement inverse pour récupérer les caractères originaux.

Exemple : compression et décompression de « abcabcabc »

Compression	String	Out	Nouvelle entrée
Initialisation			[0-255] = code ASCII
Exécution	abcabcabc	97 (= a)	258 = ab
	bcabcabc	98 (= b)	259 = bc
	cabcbabc	99 (= c)	260 = ca
	abcbabc	258	261 = abc
	cbcbabc	99 (=c)	262 = cb
	bcabc	259	263 = bca
	abc	261	
	Fin de compression		257

Décompression	In	Out	Nouvelle entrée
Initialisation			[0-255] = code ASCII
Exécution	97	a	
	98	b	258 = ab
	99	c	259 = bc
	258	ab	260 = ca
	99	c	261 = abc
	259	bc	262 = cb
	261	abc	263 = bca
	Fin de compression	257	

Taille de « abcabcabc » :

non compressé = $11 \times 8 = 88$ bits

compressé = $8 \times 10 = 80$ bits (Table indexée sur 10 bits donc 1024 entrées)

= $8 \times 9 = 72$ bits (Table indexée sur 9 bits donc 512 entrées)

figure 38 : Compression sans perte, LZW

La méthode « Run Length Encoding » (RLE) est une autre méthode de compression sans perte. Cette méthode est supportée par le format GIF. Son principe est plus simple que celui de la méthode LZW. Lorsque plusieurs pixels consécutifs ont la même valeur, au lieu de coder pour chacun leur valeur, il suffit de coder le nombre de pixels consécutifs puis leur valeur.

Exemple : *figure 1* (où w = vert, x = mauve, y = rose , z = jaune)

Forme non compressée : wxwwwwwwwwywwww

Forme compressée : 1w1x8w1y1w1z3w

Taille de *figure 1* :

non compressée = 16 pixels * 16 bits par pixels = 256 bits

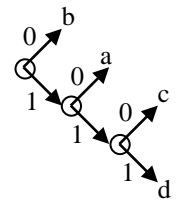
compressée = 7*(10 +16) = 182 bits (Nombre sur 10 bits donc 1024 pixels consécutifs)

figure 39 : Compression sans perte, RLE

La méthode de « Huffman » est également une méthode sans perte. Cette méthode n'est supportée par aucun format majeur. Pour compresser les données, elle les lit comme des caractères ASCII et attribue un code binaire de longueur variable en fonction des fréquences d'apparition des caractères. Ensuite, elle enregistre le code binaire correspondant aux caractères, ainsi que la table de correspondance, dans le fichier compressé. Lors de la décompression, on utilise la table afin de récupérer les caractères originaux.

Exemple : compression et décompression de « abcabdbabcb »

Compression	String	Table		
Lecture	abcabdbabcb	a	3/11	10
		b	5/11	0
		c	2/11	110
		d	1/11	111
Ecriture	a	10		
	b	100		
	c	100110		
	...	10011010011101001100		

Décompression	In		Out
	10011010011101001100		a
	011010011101001100		ab
	11010011101001100		abc
	10011101001100		abca
	011101001100		abcab
	...		abcabdbabcb

Taille de « abcabdbabcb » :

non compressé = 11*8 = 88 bits

compressé = 20 bits

figure 40 : Compression sans perte, Huffman

Annexe 2 : La méthode de compression avec perte, JPEG

La méthode de compression avec perte, la plus utilisée est, sans aucun doute, la méthode JPEG. Cette méthode est employée dans le format de fichier du même nom mais aussi dans beaucoup d'autres formats tels que TIFF et DICOM. En réalité, elle met en œuvre une transformation, « Discrete Cosine Transform » (DCT), puis une quantification lors de laquelle va avoir lieu la perte d'informations. Ces deux opérations permettent d'introduire de la redondance dans l'information numérique. Cette redondance est, par la suite, mise à profit par la méthode de « Huffman » pour effectivement compresser les données.

*Dans la transformation DCT, l'image est subdivisée en blocs de taille réduite de N*N pixels (généralement 8*8 pixels). Chaque bloc subit une transformation mathématique orthogonale inversible linéaire du domaine spatial vers le domaine fréquentiel, indépendamment des autres blocs. On obtient donc à partir d'une matrice de valeurs de pixels, une matrice de coefficients de fréquence d'apparition de ce pixel dans le bloc. Par convention, les valeurs transformées sont positionnées de telle manière que, la valeur moyenne de tous ces coefficients est placée en haut à gauche de la matrice. Plus on s'éloigne de cette position, plus la grandeur des coefficients diminue. Ce qui signifie que la transformation concentre la représentation de l'image en haut à gauche de la matrice. Pour retrouver les pixels initiaux, on applique sur les coefficients la transformation inverse. [Kaddour99]*

Transformation :
$$F(u, v) = \alpha(u) \alpha(v) \sum_{x=0}^{N-1} \sum_{y=0}^{N-1} f(x, y) \cos\left[\frac{(2x+1)u\pi}{2N}\right] \cos\left[\frac{(2y+1)v\pi}{2N}\right]$$

Transformation inverse : ~~Erreur ! Signet non défini~~
$$f(x, y) = \alpha(u) \alpha(v) \sum_{u=0}^{N-1} \sum_{v=0}^{N-1} F(u, v) \cos\left[\frac{(2x+1)u\pi}{2N}\right] \cos\left[\frac{(2y+1)v\pi}{2N}\right]$$

$x, y, u, v = 0, 1, 2, \dots, N-1$ et $N =$ la taille du bloc.
 $f(x, y)$ représente la valeur du pixel du bloc de l'image initiale aux coordonnées x et y .
 $F(u, v)$ représente le coefficient de la matrice de sortie de la DCT en u et v .

$$\alpha(z) = \begin{cases} \left(\frac{1}{N}\right)^{\frac{1}{2}} & \text{si } z = 0 \\ \left(\frac{2}{N}\right)^{\frac{1}{2}} & \text{sinon} \end{cases}$$

figure 41 : Transformation DCT

La quantification consiste à diviser chaque coefficient de la matrice obtenue lors de la DCT par son quantum correspondant dans une matrice de quantification, et ensuite, à arrondir le quotient à l'entier le plus proche. Plus les quanta sont importants, plus le nombre de coefficients réduits à 0 est élevé et plus il y a de redondances. Mais un grand diviseur signifie également que l'image subit plus de distorsions. Or, on a constaté que l'œil humain est moins sensible aux changements des couleurs les plus fréquentes qu'aux changements des couleurs qui apparaissent le moins dans une image. On utilise donc généralement une matrice de quantification dont les éléments sont petits, en haut à gauche et grands, en bas à droite. Evidemment, cette matrice doit être stockée dans le fichier compressé. Enfin, les valeurs de

cette matrice peuvent être augmentées ou diminuées, en ajustant le facteur de qualité.
[Watson95]

Quantification : $F'(u, v) = \text{round} (F(u, v) / q(u, v))$

'Dé-Quantification' : $F''(u, v) = F'(u, v) * q(u,v) = F(u, v) + \text{error}(u,v)$

figure 42 : Quantification et Dé-quantification

Annexe 3 : Les types de câble pour la transmission de données

Le câble électrique à paires torsadées est, sans doute, le dispositif de transmission dont l'installation est la plus simple et dont l'utilisation est la plus courante. Il est composé de paires de fils en cuivre torsadés l'un autour de l'autre. L'entièreté du câble est entourée d'une gaine en plastique dur. Ce media est normalement dédié au transport de données dont le débit maximal varie entre 1 Mbps et 100 Mbps. Il existe cinq catégories de câbles à paires torsadées qui certifient chacune le transfert de données à un débit maximal de, respectivement, 1 Mbps, 4Mbps, 10 Mbps, 16Mbps et 100Mbps. Les câbles des catégories un, deux et quatre, sont également munis d'une protection contre les interférences électromagnétiques, cependant ce blindage augmente l'atténuation du signal et d'autres artefacts qui peuvent mener à la perte de données. Enfin, les câbles à paires torsadées peuvent être utilisés par segmentation de quelques centaines de mètres dont le coût est assez faible.

Un câble électrique, appelé câble coaxial, est également utilisé comme support de transmission. Ce second type de câbles est composé d'un conducteur central rigide en cuivre, entouré d'un matériau isolant, enveloppé à son tour par une tresse conductrice en cuivre. L'ensemble est protégé par une gaine isolante en matière plastique. Le câble coaxial permet un meilleur blindage contre les interférences et une utilisation sur des portions plus longues. Cependant la bande passante du câble coaxial dépend de sa longueur et de son diamètre. Il est toutefois possible d'atteindre un débit de l'ordre de 1 Gbps sur une longueur d'un kilomètre.

Le câble à fibre optique regroupe une ou plusieurs fibres optiques qui sont des médias ténus et flexibles, formés d'un cœur en verre, à l'intérieur duquel se propagent les données sous forme d'impulsions lumineuses. Le cœur est enrobé d'une gaine en verre présentant un indice de réfraction inférieur de façon à concentrer les impulsions à l'intérieur du cœur. L'ensemble est abrité au sein d'une gaine en matière plastique. Grâce à ce type de média, on peut accéder à un débit supérieur à 4 Gbps sur des distances de l'ordre de 10 kilomètres. Evidemment de telles performances ne peuvent être atteintes qu'à l'encontre de certains inconvénients tels que la difficulté d'installation et le coût de ce type de support.

Annexe 4 : Les normes d'implémentation pour les réseaux LAN

Ethernet

L'Ethernet est l'une des plus vieilles, plus simples et moins onéreuses technologies adaptées aux réseaux LAN, dont la topologie s'inscrit sous forme de bus. Cette norme peut être utilisée avec plusieurs types de médias et définit son propre protocole (MAC802.3), qui est d'ailleurs très proche de CSMA. Elle fonctionne à un débit de 10 Mbps.

Nom	Type du câble	Longueur max. d'un segment	Nombre max. de stations
10Base5	Coaxial épais	500 m	100
10Base2	Coaxial fin	200 m	30
10Base-T	Paires torsadées	100 m	1024
10Base-F	Fibre optique	2000 m	1024

figure 43 : Types les plus fréquents d'ethernet

Ce débit de 10 Mbps est théorique car dans un Ethernet, plus il y a de machines connectées plus le niveau de bande passante disponible est faible. La solution à cette limitation est l'Ethernet commuté. Le cœur de ce type de réseaux est un commutateur disposant d'un bus interne à très haut débit, typiquement 1Gbps et d'un ensemble de connecteurs permettant la connexion des machines. Dès qu'une machine connectée transmet des données, le commutateur détermine le connecteur sur lequel la machine destinataire est connectée et envoie les données uniquement sur ce connecteur. Etant donné que le commutateur respecte les spécifications fonctionnelles de la norme, tous les problèmes d'accès sont gérés. Cette architecture de réseau permet d'assurer le débit de 10 Mbps quel que soit le nombre de machines.

Token Ring

Token Ring est une seconde norme d'implémentation de réseaux LAN, mais cette fois, l'architecture est sous forme d'anneau. Cette norme recommande l'utilisation de câbles à paires torsadées blindées et définit un débit de 1, 4 ou 16 Mbps sur des segments d'environ 100 mètres . Elle possède également son propre protocole (MAC 802.5).

Fast Ethernet

La norme de réseaux Fast Ethernet est une adaptation de l'Ethernet à un débit supérieur. Elle est capable de transmettre des données à un débit de 100 Mbps. Le Fast Ethernet supporte comme son cousin l'Ethernet plusieurs types de médias.

Nom	Type du câble	Longueur max. d'un segment
100Base-T4	Paires torsadées (3 à 4 paires de catégorie 3)	100 m
100Base-TX	Paires torsadées (2 paires de catégorie 5)	100 m
100Base-FX	Fibre optique	2000 m

figure 44 : Types les plus fréquents de fast ethernet

Le niveau de bande passante disponible avec le Fast-Ethernet subit également une diminution en fonction du nombre de machines connectées. La solution est identique que pour l'Ethernet, il est possible d'établir un Fast-Ethernet commuté.

Fiber Distributed Data Interface FDDI

Le réseau FDDI (Fiber Distributed Data Interface) est un réseau LAN en anneau en fibre optique à hautes performances. Il offre un débit de 100 Mbps et permet des distances de raccordement pouvant atteindre 200 km entre un nombre maximum de 1000 stations. Le réseau FDDI comporte son propre protocole dont les bases sont très proches de celles du Token Ring. Les inconvénients de ce type de réseaux sont la complexité des composants mis en œuvre et le coût qu'ils génèrent.

Giga Ethernet

La norme Giga Ethernet est la suite logique de l'évolution des normes Ethernet et Fast Ethernet. Cette norme est capable de transmettre des données à un débit de 1000 Mbps. Le Giga Ethernet supporte plusieurs types de médias.

Nom	Type du câble	Longueur max. d'un segment
1000Base-SX	Fibre optique (multimode, longueur d'onde 770-860 nm)	220-550 m
1000Base-LX	Fibre optique (monomode ou multimode, longueur d'onde 1270-1355 nm)	550 m (multimode) 5000 m (monomode)
1000Base-CX	Paires torsadées (2 paires de catégorie 5)	25 m

figure 45 : Types les plus fréquents de giga ethernet

Annexe 5 : Les réseaux publics

Le RTC (Réseau téléphonique commuté)

Les réseaux téléphoniques sont structurés de façon hiérarchique. Des zones primaires recouvrant une surface géographique importante, sont divisées en zones secondaires toutes connectées à un commutateur primaire. Ces zones secondaires sont également séparées en zones plus petites reliées à un commutateur secondaire. Ces dernières disposent d'un centre de rattachement auquel les abonnés sont raccordés soit directement, soit par l'intermédiaire d'un centre téléphonique local. Aujourd'hui les réseaux téléphoniques sont numérisés dans leur quasi totalité. Cependant les liaisons entre les abonnés et les centres de rattachement ou les centres locaux, appelées boucles locales, sont encore analogiques. Par conséquent, lorsque l'on veut utiliser les réseaux téléphoniques comme moyen de communication pour un ordinateur, il est nécessaire de convertir les signaux binaires issus de l'ordinateur en signaux analogiques transmissibles sur la boucle locale. Cette tâche est réalisée à l'aide d'un modem. Généralement, les modems permettent des débits maximum de 33.3 Kbps ou 56 Kbps. Les réseaux téléphoniques fournissent la méthode la moins chère mais aussi la plus lente d'interconnexion.

L'ADSL(Asymmetric Digital Subscriber Line)

L'ADSL est une technologie de communication relativement nouvelle à la fois fiable et robuste qui permet, sur le réseau téléphonique existant, de transmettre des signaux digitaux. Comme son nom l'indique, cette technologie est asymétrique, ce qui signifie que la vitesse de transmission dans un sens n'est pas la même que dans l'autre. Les vitesses de transmission atteintes sont de l'ordre de 512 Kbps pour le flux de données ascendant et de 1 Mbps pour le flux descendant. Cette caractéristique fait d'ADSL une technologie idéale pour tous les types d'accès où l'utilisateur reçoit en général bien plus d'informations qu'il n'en renvoie. Pour compléter un circuit ADSL, il est nécessaire de disposer, du côté utilisateur, d'un modem ADSL et d'un coupleur de téléphonie analogique. Le modem ADSL module et démodule les données numériques en utilisant des algorithmes complexes, de sorte qu'elles puissent être transmises.

Le coupleur est utilisé pour combiner les services téléphoniques analogiques et les services ADSL numériques, permettant ainsi aux services téléphoniques traditionnels de coexister avec les nouveaux services rapides. D'une manière générale, le modem ADSL et le coupleur sont regroupés au sein d'un même boîtier "le modem ADSL".

L'ISDN (Integrated Services Digital Network)

Les opérateurs de télécommunication ont mené des études afin de développer un réseau de communication numérique de bout en bout. Ces études ont abouti aux réseaux numériques à intégration de services (ISDN) qui offrent un moyen plus performant et mieux adapté, pour interconnecter à distance des ordinateurs entre eux. L'ISDN offre deux types d'accès.

L'accès de base compte deux canaux de type B à 64 Kbps chacun et un canal de contrôle de type D à 16 Kbps. Ce qui fournit un débit de 128 Kbps.

L'accès primaire combine vingt-trois (pour les USA) ou trente (pour l'Europe) canaux de type B et un canal de type D à 64 Kbps, qui procurent un débit de 1544Mbps ou 2048 Mbps.

Cette technologie nécessite cependant l'installation d'un minimum d'équipements de la part de l'utilisateur. Une terminaison numérique de réseau, reliée au centre de raccordement ISDN, est installée par l'opérateur, à l'extrémité de la liaison numérique. L'abonné raccorde son

téléphone ou ses terminaux à cette terminaison numérique, soit directement, soit par l'intermédiaire d'une terminaison numérique d'abonné, qui joue le rôle de commutateur local. Il faut également ajouter un adaptateur entre la terminaison et les terminaux lorsque ceux-ci ne sont pas compatibles à l'ISDN.

Le câble

Le réseau de télédistribution peut également fournir un moyen de communication efficace. Cette méthode d'accès nécessite la mise en place d'un modem particulier et permet généralement d'atteindre des débits de 3 à 10 Mbps dans le sens descendant et 200 Kbps à 2 Mbps dans le sens ascendant.

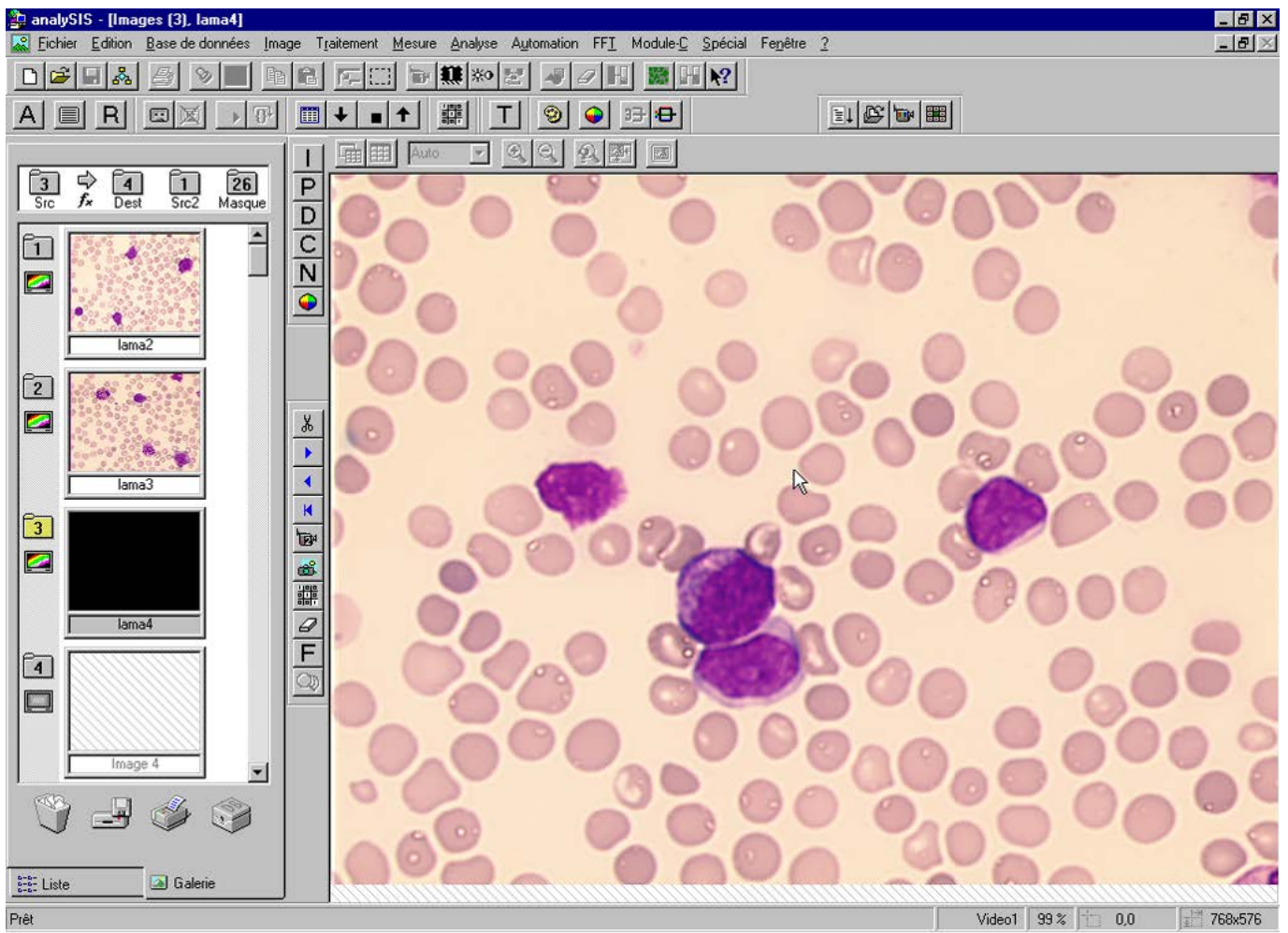
L'ATM (Asynchronous Transfert Mode)

Le mode de transfert asynchrone (ATM) est une technologie de réseau qui est capable de livrer un débit de 25 à 622 Mbps. L'ISDN à large bande (B-ISDN) emploie le protocole ATM dans l'une de ses différentes couches. Cette technologie permet de transmettre les données dans des paquets de taille fixe à des vitesses déterminées par la nature des données et le type de service souscrit par l'utilisateur. Cependant, elle est encore très chère et peu utilisée.

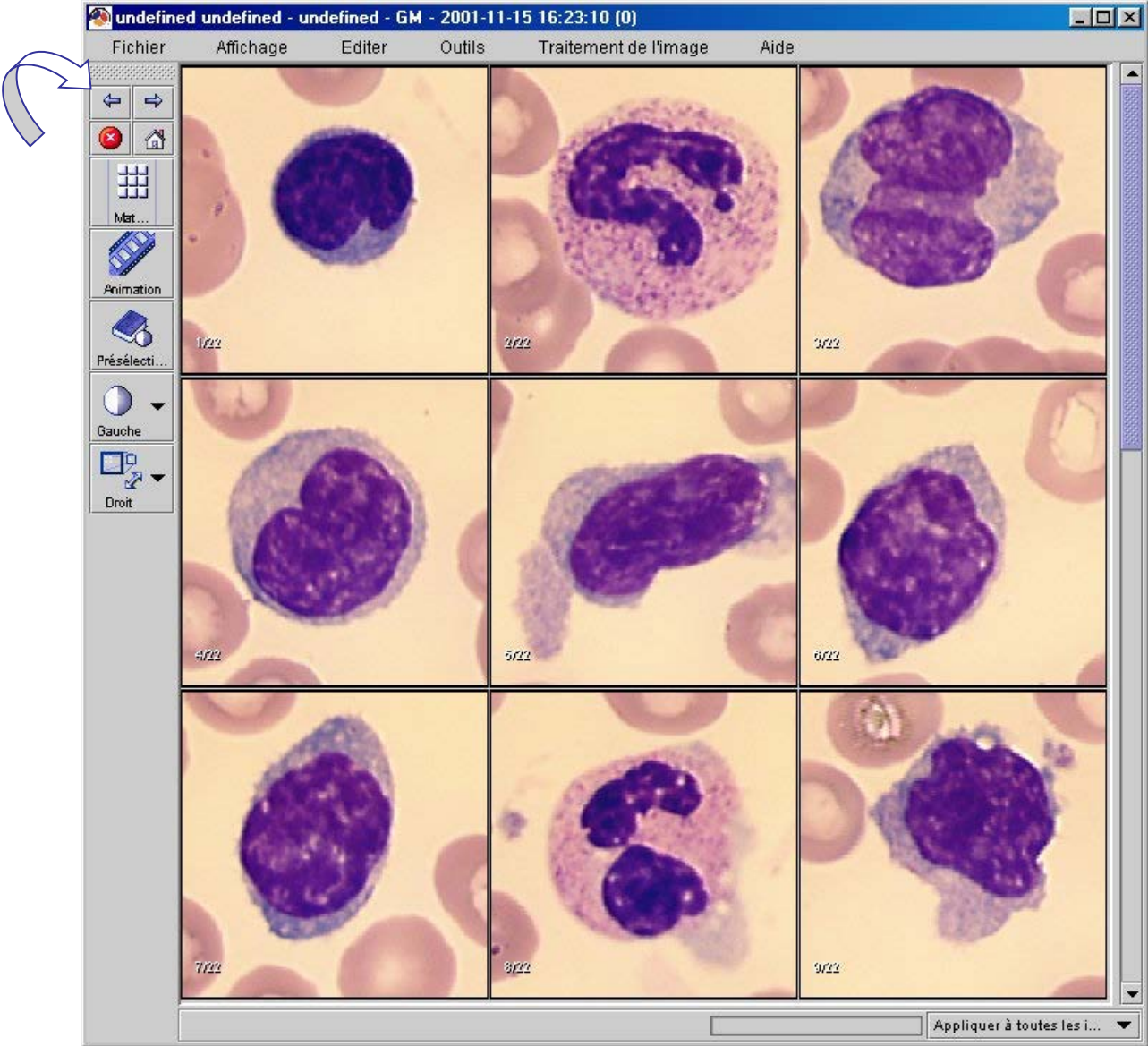
Type d'accès	RTC	ISDN (de base)	ADSL	Cable
Débit théorique (récept./émis.)	56Kbps /33,6Kbps	128Kbps /128Kbps	1Mbps/512Kbps	1Mbps à 512Kbps /512Kbps à 128Kbps
Type de facturation	A la durée	A la durée	Forfaitaire	Forfaitaire
Matériel nécessaire	Modem (à partir de 49,58 EUR)	Adaptateur (± 111,55 EUR)	Modem ADSL (de 136,09 EUR à 297,22 EUR ou location de 7,50 EUR par mois)	Modem câble (± 266,49 EUR ou location de 10,34 EUR par mois)
Frais d'installation	Aucun	101,98 EUR	De 24,54 EUR à 210,71 EUR	De 49,55 EUR à 198,29 EUR
Abonnement mensuel	16,19 EUR	36,29 EUR	16,19 EUR	A partir de 35,99 EUR
Volume d'émission	Illimité	Illimité	Illimité	Emission limitée
Abonnement mensuel au fournisseur d'accès à Internet	0 à 14,87 EUR et plus	0 à 14,87 EUR et plus	39,54 EUR à 54,41 EUR	0 EUR

figure 46 : Tableau des prix approximatifs au mois de décembre 2001 en Belgique [CRID00]

Annexe 6 : L'interface de « Analysis »



Annexe 7 : Les interfaces de « Télémis »



Annexe 8 : L'élaboration d'un format compatible avec TIFF et DICOM

Format TIFF

Un fichier au format TIFF est structuré séquentiellement par un ensemble de bytes (8bits) qui peuvent être numérotés de 0 à N. Un fichier au format TIFF est toujours composé d'une en-tête et d'un certain nombre de fragments (au moins un), appelés « Image File Directory » (IFD). Chaque IFD est lui-même composé de plusieurs sections (au moins une), appelées « Directory Entry ».

La structure d'un fichier TIFF commence toujours par l'en-tête de 8 bytes numérotés de 0 à 7 dont les deux premiers bytes (0 à 1) déterminent l'ordre des bytes utilisés dans le fichier (« II » pour little-endian et « MM » pour big-endian). Les deux bytes suivants (2 à 3) contiennent un « magic number » (« 42 ») qui est utilisé pour identifier le fichier comme étant bien au format TIFF. Les quatre derniers bytes établissent le lien vers le premier IFD en spécifiant le numéro séquentiel de son premier byte.

Par contre, un IFD peut être localisé n'importe où dans le fichier et est constitué de $2+B*12+4$ bytes (B est le nombre d'entrées). Les deux premiers bytes (A à A+1) de l'IFD sont utilisés pour fixer le nombre d'entrées. Ensuite les $B*12$ bytes suivants (A+2 à A+2+B*12-1) sont consacrés aux entrées. Enfin, les quatre derniers bytes précisent le numéro séquentiel de l'IFD suivant.

Les entrées sont composées de 12 bytes. Les deux premiers bytes identifient l'entrée par un « Tag », nombre particulier qui détermine la sémantique des valeurs stockées dans l'entrée. Les deux bytes suivants spécifient le type des valeurs de l'entrée (par exemple : 3=SHORT [16-bits unsigned integer], 4=LONG [32-bit unsigned integer], 5=RATIONAL [2-LONG numérateur puis dénominateur], etc.). Les quatre bytes suivants de l'entrée sont destinés à fournir le nombre de valeurs que contient l'entrée. Et enfin, les quatre derniers bytes de l'entrée donnent soit les valeurs, si elles peuvent être codées sur 4 bytes, soit le numéro du premier byte de la section où elles se trouvent.

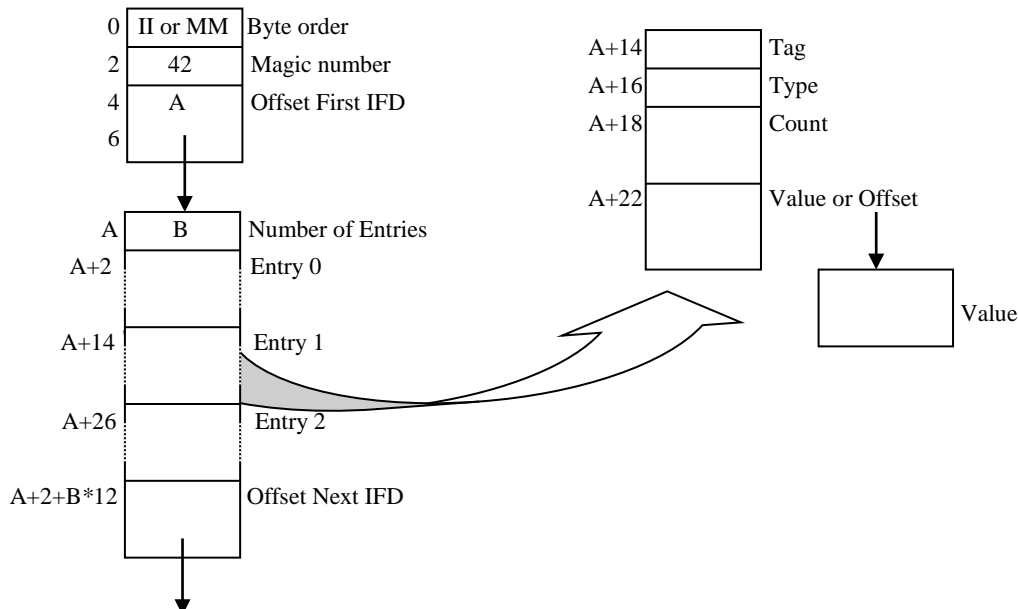


figure 47 : Organisation physique des informations sous format TIFF

Un fichier au format TIFF est donc composé d'entrées. Certaines d'entre elles sont obligatoires tandis que d'autres sont facultatives. Parmi les entrées facultatives, on retrouve

celles définies par le standard mais également n'importe quelle entrée « propriétaire » ajoutée à la structure pourvu qu'elle respecte l'organisation décrite plus haut. En ce qui concerne les entrées obligatoires, cela dépend du type d'image enregistrée dans le fichier.

Par exemple, pour les images RGB les entrées obligatoires sont les suivantes :

TagName	Decimal	Hex	Type	Value
ImageWidth	256	100	SHORT or LONG	
ImageLength	257	101	SHORT or LONG	
BitsPerSample	258	102	SHORT	8,8,8
Compression	259	103	SHORT	1 or 2
PhotometricInterpretation	262	106	SHORT	2
StripOffsets	273	111	SHORT or LONG	
SamplesPerPixel	277	115	SHORT	3 or more
RowsPerStrip	278	116	SHORT or LONG	
StripByteCounts	279	117	SHORT or LONG	
Xresolution	282	11A	RATIONAL	
Yresolution	283	11B	RATIONAL	
ResolutionUnit	296	128	SHORT	1, 2 or 3

figure 48 : Entrées obligatoires pour une image RGB sous format TIFF

L'entrée « ImageWidth » permet de définir le nombre de lignes dans l'image.

L'entrée « ImageLength » permet de définir le nombre de colonnes dans l'image.

L'entrée « SamplesPerPixel » permet de définir le nombre de composants par pixel
(3 : Rouge, Vert, Bleu ou plus)

L'entrée « BitsPerSample » permet de définir le nombre de bits par composant.
(8-bits pour le Rouge, 8-bits pour le Vert, 8-bits pour le Bleu)

L'entrée « Compression » permet de définir le type de compression des données :

1 = Pas de compression

2 = Compression Huffman

L'entrée « PhotometricInterpretation » permet de définir le type d'image.

2 = RGB

Les données (image) peuvent être enregistrées partout dans le fichier. Elles peuvent également être séparées en plusieurs parties.

L'entrée « RowsPerStrip » permet de définir le nombre de lignes dans chaque partie.

(Exemple : S'il y a 24 lignes et que l'on définit 10 lignes par partie, alors il y aura 3 parties : 2 de 10 lignes et une de 4 lignes)

L'entrée « StripOffsets » permet de définir l'offset de chaque partie.

L'entrée « StripByteCounts » permet de définir le nombre de bytes dans chaque partie.

L'entrée « ResolutionUnit » permet de définir le type d'unité de résolution

1 = aucune mesure particulière

2 = Inch

3 = Centimètre

L'entrée « Xresolution » est le nombre de pixels par unité de résolution dans les lignes

L'entrée « Yresolution » est le nombre de pixels par unité de résolution dans les colonnes

L'entrée « PlanarConfiguration » permet de définir la manière dont les composants des pixels sont encodés. Cette entrée est facultative car elle est par défaut à la valeur 1.

- 1 = Chunky format (RGBRGBRGBRGB)
- 2 = Planar format (RRRRGGGGBBBB)

Format compatible avec TIFF et DICOM

Une première stratégie de combinaison consisterait à encapsuler l'header ainsi qu'un seul IFD contenant toutes les entrées du format TIFF dans le préambule de 128 bytes du format de fichier de la partie 10 de DICOM. Les entrées, qui ont pour valeur un offset, définiraient comme valeur un offset pointant vers une valeur encapsulée dans le Data Set. Cependant cette stratégie restreint le nombre d'entrées à 9 car le préambule ne fournit que 128 bytes dont 8 sont utilisés pour l'header et 6 pour l'IFD, ce qui ne laisse que 114 bytes pour 9 entrées de 12 bytes. Or pour une image RGB, on compte déjà 12 entrées obligatoires.

Une deuxième stratégie de combinaison consisterait à encapsuler uniquement l'header du format TIFF dans le préambule et de faire pointer l'offset de l'header vers l'IFD qui serait lui-même encapsulé dans un Data Element privé du Data Set, (par exemple, (0041,0000) IFD TIFF) ou dans un Data Element standard mais non utilisé et dont la taille est suffisante (par exemple, (0020,4000) Image Comments qui a une taille maximum de 10240 bytes ou (0002,0102) Private Information qui n'a pas de limite de taille). Bien entendu, pour ne pas dupliquer les valeurs des pixels, il suffit de faire pointer la valeur de l'entrée « StripOffsets » de l'IFD encapsulée vers les valeurs des pixels du Data Element « Pixel Data » du format détaillé dans le standard DICOM.

Cette deuxième stratégie est théoriquement valable et permet d'utiliser un format compatible aux deux standards mais dans la pratique, on a observé que plusieurs logiciels gérant le format compris dans le standard DICOM ne parviennent plus à traiter les fichiers composés selon cette stratégie. Ce phénomène s'explique par le fait que la spécification du standard impose que les 128 bytes du préambule soient placés à la valeur 0 s'ils ne sont pas utilisés. Par extension, les auteurs de certains logiciels ont considéré que le préambule commence toujours par 128 bytes mis à la valeur 0. Dès lors, dès qu'un fichier ne commence pas par 128 bytes mis à cette valeur, il est considéré comme ne respectant pas le format spécifié dans le standard DICOM. En conclusion, il n'est pas possible d'utiliser ce format compatible sans devoir modifier certains logiciels.

Annexe 9 : Les DTD du modèle de document multimédia médical

DTD correspondant au niveau 1

```
<?xml version="1.0" encoding="UTF-8"?>
<!-- DTD For Dossier Medical Multimedia -->

<!ELEMENT DMM (Type_DMM, Titre_DMM, Volume_DMM, Nom_DMM, Version_DMM, Statut_DMM, Origine_DMM,
Date_Creation_DMM, Date_Transmission_DMM, Date_Validation_DMM, Patient, Examen*)>
<!-- Attributs du DMM -->
<!ELEMENT Type_DMM (#PCDATA)>
<!ELEMENT Titre_DMM (#PCDATA)>
<!ELEMENT Volume_DMM (#PCDATA)>
<!ELEMENT Nom_DMM (#PCDATA)>
<!ELEMENT Version_DMM (#PCDATA)>
<!ELEMENT Statut_DMM (#PCDATA)>
<!ELEMENT Origine_DMM (#PCDATA)>
<!ELEMENT Date_Creation_DMM (#PCDATA)>
<!ELEMENT Date_Transmission_DMM (#PCDATA)>
<!ELEMENT Date_Validation_DMM (#PCDATA)>

<!ELEMENT Patient (Numero_Patient, Nom_Patient, Prenom_Patient, Sexe_Patient?, Nom_Jeune_Fille_Patient?, Taille_Patient?,
Poids_Patient?, Date_Naissance_Patient, Adresse_Patient?, Telephone1_Patient?, Telephone2_Patient?, Mutuel_Patient?,
Coordonnees_Facturation_Patient?, Commentaire_Patient?, Admission?, Medecin?)>
<!-- Attributs du patient -->
<!ELEMENT Numero_Patient (#PCDATA)>
<!ELEMENT Nom_Patient (#PCDATA)>
<!ELEMENT Prenom_Patient (#PCDATA)>
<!ELEMENT Sexe_Patient (#PCDATA)>
<!ELEMENT Nom_Jeune_Fille_Patient (#PCDATA)>
<!ELEMENT Taille_Patient (#PCDATA)>
<!ELEMENT Poids_Patient (#PCDATA)>
<!ELEMENT Date_Naissance_Patient (#PCDATA)>
<!ELEMENT Adresse_Patient (#PCDATA)>
<!ELEMENT Telephone1_Patient (#PCDATA)>
<!ELEMENT Telephone2_Patient (#PCDATA)>
<!ELEMENT Mutuel_Patient (#PCDATA)>
<!ELEMENT Coordonnees_Facturation_Patient (#PCDATA)>
<!ELEMENT Commentaire_Patient (#PCDATA)>

<!ELEMENT Admission (Heure_Admission, Date_Admission, Nom_Institution_Admission, Adresse_Institution_Admission?,
Departement_Admission, Description_Admission?)>
<!-- Attributs de l'admission -->
<!ELEMENT Heure_Admission (#PCDATA)>
<!ELEMENT Date_Admission (#PCDATA)>
<!ELEMENT Nom_Institution_Admission (#PCDATA)>
<!ELEMENT Adresse_Institution_Admission (#PCDATA)>
<!ELEMENT Departement_Admission (#PCDATA)>
<!ELEMENT Description_Admission (#PCDATA)>

<!ELEMENT Medecin (Nom_Medecin, Prenom_Medecin, Numero_Medecin, Adresse_Medecin?)>
<!ELEMENT Nom_Medecin (#PCDATA)>
<!ELEMENT Prenom_Medecin (#PCDATA)>
<!ELEMENT Numero_Medecin (#PCDATA)>
<!ELEMENT Adresse_Medecin (#PCDATA)>

<!ELEMENT Examen (Type_EMM, Titre_EMM, Volume_EMM, Nom_EMM, Version_EMM, Statut_EMM, Emplacement_EMM)>
<!-- Attributs des examens -->
<!ELEMENT Type_EMM (#PCDATA)>
<!ELEMENT Titre_EMM (#PCDATA)>
<!ELEMENT Volume_EMM (#PCDATA)>
<!ELEMENT Nom_EMM (#PCDATA)>
<!ELEMENT Version_EMM (#PCDATA)>
<!ELEMENT Statut_EMM (#PCDATA)>
<!ATTLIST Examen Text (false | true) 'false'
Image (false | true) 'false'
Audio (false | true) 'false'
Video (false | true) 'false'>

<!ELEMENT Emplacement_EMM (#PCDATA)>
<!ATTLIST Emplacement_EMM xml:link CDATA #FIXED 'SIMPLE'
href CDATA #REQUIRED>
```

DTD correspondant au niveau 2

```
<?xml version="1.0" encoding="UTF-8"?>
<!-- DTD pour Examen Medical Multimedia en Cyto-Hematologie -->

<!ELEMENT EMMCH (Type_EMM, Titre_EMM, Volume_EMM, Nom_EMM, Version_EMM, Statut_EMM, Emplacement_EMM,
Presentation_EMM*, Diagnostic, Commentaire, Enregistrement, Contexte?, Prescripteur*, Observateur*, Media*)>
<!-- Attributs de l'examen -->
<!ELEMENT Type_EMM (#PCDATA)>
<!ELEMENT Titre_EMM (#PCDATA)>
<!ELEMENT Volume_EMM (#PCDATA)>
<!ELEMENT Nom_EMM (#PCDATA)>
<!ELEMENT Version_EMM (#PCDATA)>
<!ELEMENT Statut_EMM (#PCDATA)>
<!ATTLIST EMMCH Text (false | true) 'false'
                Image (false | true) 'false'
                Audio (false | true) 'false'
                Video (false | true) 'false'

<!ELEMENT Emplacement_EMM (#PCDATA)>
<!ATTLIST Emplacement_EMM xml:link CDATA #FIXED 'SIMPLE'
                href CDATA #REQUIRED>

<!ELEMENT Presentation_EMM (Acces, Emplacement_Presentation)>
<!ELEMENT Acces (#PCDATA)>
<!ELEMENT Emplacement_Presentation (#PCDATA)>

<!ELEMENT Diagnostic (#PCDATA)>
<!ELEMENT Commentaire (#PCDATA)>

<!ELEMENT Enregistrement (Numero_Enregistrement, Heure_Enregistrement?, Date_Enregistrement?,
Nom_Institution_Enregistrement?, Adresse_Institution_Enregistrement?)>
<!-- Attributs de l'enregistrement -->
<!ELEMENT Numero_Enregistrement (#PCDATA)>
<!ELEMENT Heure_Enregistrement (#PCDATA)>
<!ELEMENT Date_Enregistrement (#PCDATA)>
<!ELEMENT Nom_Institution_Enregistrement (#PCDATA)>
<!ELEMENT Adresse_Institution_Enregistrement (#PCDATA)>

<!ELEMENT Contexte (Heure_prelevement?, Date_Prelevement, Mode_Prelevement, Nature_Prelevement, Site_prelevement,
Contexte_Clinique?, Recherche_Particieliere?, Indication_Particieliere?)>
<!-- Attributs du contexte -->
<!ELEMENT Heure_prelevement (#PCDATA)>
<!ELEMENT Date_prelevement (#PCDATA)>
<!ELEMENT Mode_prelevement (#PCDATA)>
<!ELEMENT Nature_prelevement (#PCDATA)>
<!ELEMENT Site_prelevement (#PCDATA)>
<!ELEMENT Contexte_Clinique (#PCDATA)>
<!ELEMENT Recherche_Particieliere (#PCDATA)>
<!ELEMENT Indication_Particieliere (#PCDATA)>

<!ELEMENT Prescripteur(Nom_Prescripteur, Prenom_Prescripteur, Numero_Prescripteur, Adresse_Prescripteur?)>
<!-- Attributs du prescripteur -->
<!ELEMENT Nom_Prescripteur (#PCDATA)>
<!ELEMENT Prenom_Prescripteur (#PCDATA)>
<!ELEMENT Numero_Prescripteur (#PCDATA)>
<!ELEMENT Adresse_Prescripteur (#PCDATA)>

<!ELEMENT Observateur (Nom_Observateur, Prenom_Observateur, Numero_Observateur, Adresse_Observateur?, Observation+)>
<!-- Attributs de l'observateur -->
<!ELEMENT Nom_Observateur (#PCDATA)>
<!ELEMENT Prenom_Observateur (#PCDATA)>
<!ELEMENT Numero_Observateur (#PCDATA)>
<!ELEMENT Adresse_Observateur (#PCDATA)>

<!ELEMENT Observation (Description_Lesion, Difficultes_Rencontrees?, Commentaire?, Media)>
<!-- Attributs du Observation -->
<!ELEMENT Description_Lesion (#PCDATA)>
<!ELEMENT Difficultes_Rencontrees (#PCDATA)>
<!ELEMENT Commentaire (#PCDATA)>

<!ELEMENT Media (Type_Media, Volume_Media, Nom_Media, Version_Media, Emplacement_Media, Heure_Acquisition?,
Date_Acquisition?, Grossissement?)>
<!-- Attributs du media -->
<!ELEMENT Type_Media (#PCDATA)>
```

```

<!ELEMENT Volume_Media (#PCDATA)>
<!ELEMENT Nom_Media (#PCDATA)>
<!ELEMENT Version_Media (#PCDATA)>
<!ELEMENT Emplacement_Media (#PCDATA)>
<!ATTLIST Emplacement_Media xml:link CDATA #FIXED 'SIMPLE'
                        href CDATA #REQUIRED>
<!ELEMENT Heure_Acquisition (#PCDATA)>
<!ELEMENT Date_Acquisition (#PCDATA)>
<!ELEMENT Grossissement (#PCDATA)>

```

DTD correspondant au niveau 3

```

<?xml version="1.0" encoding="UTF-8"?>
<!-- DTD pour Examen Medical Multimedia en Cyto-Hematologie -->

<!ELEMENT Scenario (Acces, Emplacement_Scenario, Conversion+, Transformation+)>
<!ELEMENT Acces (#PCDATA)>
<!ELEMENT Emplacement_Scenario (#PCDATA)>
<!ELEMENT Conversion (Type_In, Type_OutFeuille, Compression)>
<!ELEMENT Type_In (#PCDATA)>
<!ELEMENT Type_Out (#PCDATA)>
<!ELEMENT Compression (#PCDATA)>
<!ELEMENT Transformation (Input+, Feuille, Output)>
<!ELEMENT Input (#PCDATA)>
<!ATTLIST Input type CDATA #REQUIRED
                href CDATA #REQUIRED>
<!ELEMENT Feuille (#PCDATA)>
<!ATTLIST Feuille type CDATA #REQUIRED
                href CDATA #REQUIRED>
<!ELEMENT Output (#PCDATA)>
<!ATTLIST Output type CDATA #REQUIRED
                href CDATA #REQUIRED>

```

Annexe 10 : Les scénarios réservés à l'examen de cytologie hématologique

Scénario réservé à la transmission vers l'ADICAP

- Le fichier XML de coordination de ce scénario est le suivant :

```
<Scenario>
  <Acces>html</Acces>
  <Emplacement_Presentation>Scenario.xml</Emplacement_Presentation>
  <Transformation>
    <Input type="primaire" href="DMM.xml">DMM.xml</Input>
    <Feuille type="xsl" href="DMMhtm.xml">DMMhtm.xml</Feuille>
    <Output type="primaire" href="Numero.htm">Numero.htm</Output>
  </Transformation>
  <Transformation>
    <Input type="primaire" href="EMMCH.xml">EMMCH.xml</Input>
    <Feuille type="xsl" href="EMMCHhtm.xml">EMMCHhtm.xml</Feuille>
    <Output type="secondaire" href="Examen1.htm">Examen1.htm</Output>
  </Transformation>
  <Transformation>
    <Input type="primaire" href="EMMCH.xml">EMMCH.xml</Input>
    <Feuille type="xsl" href="FichierIni.xml">FichierIni.xml</Feuille>
    <Output type="secondaire" href="Numero.ini">Numero.ini</Output>
  </Transformation>
  <Conversion_Media>
    <Type_Media_In>Image/Tif</Type_Media_In>
    <Type_Media_Out>Image/Jpg</Type_Media_Out>
    <Compression>55</Compression>
  </Conversion_Media>
</Scenario>
```

On remarque que ce scénario a comme identificateur (attribut « Acces ») la valeur « html ». Ce scénario est composé de trois transformations, toutes de type XSLT, car l'attribut « type » de l'élément « Feuille » de toutes ces transformations possède la valeur « xsl ».

La première transformation utilise la feuille de style « DMMhtm.xml » pour transformer le fichier source « DMM.xml » qui respecte la DTD correspondant au niveau 1, c'est-à-dire regroupant les informations associées à tout le document multimédia. On obtient alors un fichier cible nommé « Numero.htm ».

La deuxième transformation utilise la feuille de style « EMMCHhtm.xml » afin de transformer le fichier source « EMMCH.xml » qui respecte la DTD correspondant au niveau 2, c'est-à-dire regroupant les informations associées à l'examen en cytologie hématologique. On obtient le fichier « Examen1.htm ».

La troisième transformation utilise la feuille de style « FichierIni.xml » afin de transformer le fichier source « EMMCH.xml » qui respecte la DTD correspondant au niveau 2, c'est-à-dire regroupant les informations associées à l'examen en cytologie hématologique. On obtient le fichier « Numero.ini ».

Le scénario est également constitué d'une conversion de média qui convertit tous les médias dont le type est « Image/Tif » en média de type « Image/Jpg » avec un degré de compression de 55. En d'autres termes toutes les images TIFF sont converties en images JPEG compressées.

- La feuille de style « DMMhtm.xml » est la suivante :

```

<?xml version="1.0" encoding="UTF-8"?>
<xsl:stylesheet version="1.0" xmlns:xsl="http://www.w3.org/1999/XSL/Transform">
  <xsl:template match="DMM">
    <html>
      <body>
        <h1 align="center"><xsl:apply-templates select="Origine_DMM"/></h1>
        <h2 align="center">
          <font color="#0080FF"><xsl:apply-templates select="Titre_DMM"/></font>
        </h2>
        <h3 align="center">
          <font color="#0080FF"><xsl:apply-templates select="Date_Creation_DMM"/></font>
        </h3>
        <hr></hr>
        <h4 align="left"><u>Informations Generales</u></h4>
        <xsl:apply-templates select="Patient"/>
        <hr></hr>
        <h4><u>Les examens</u></h4>
        <ul type="square">
          <xsl:apply-templates select="Examen"/>
        </ul>
      </body>
    </html>
  </xsl:template>

  <xsl:template match="Patient">
    <ul type="square">
      <li><p align="left"><u>Patient</u></p></li>
      <ul type="disc">
        <li>Numero : <xsl:apply-templates select="Numero_Patient"/></li>
        <li>Nom : <xsl:apply-templates select="Nom_Patient"/></li>
        <li>Prenom : <xsl:apply-templates select="Prenom_Patient"/></li>
        <li>Sexe : <xsl:apply-templates select="Sexe_Patient"/></li>
        <li>Nom de jeune fille : <xsl:apply-templates select="Nom_Jeune_Fille_Patient"/></li>
        <li>Taille : <xsl:apply-templates select="Taille_Patient"/></li>
        <li>Poids : <xsl:apply-templates select="Poids_Patient"/></li>
        <li>Date de naissance : <xsl:apply-templates select="Date_Naissance_Patient"/></li>
        <li>Adresse : <xsl:apply-templates select="Adresse_Patient"/></li>
        <li>Telephone1 : <xsl:apply-templates select="Telephone1_Patient"/></li>
        <li>Telephone2 : <xsl:apply-templates select="Telephone2_Patient"/></li>
        <li>Mutuel : <xsl:apply-templates select="Mutuel_Patient"/></li>
        <li>Coordonnees de facturation:<xsl:apply-templates select="Coordonnees_Facturation_Patient"/></li>
        <li>Commentaire : <xsl:apply-templates select="Commentaire_Patient"/></li>
      </ul>
    </ul>
    <xsl:apply-templates select="Admission"/>
    <xsl:apply-templates select="Medecin"/>
  </xsl:template>

  <xsl:template match="Admission">
    <ul type="square">
      <li><p align="left"><u>Admission</u></p></li>
      <ul type="disc">
        <li>Heure : <xsl:apply-templates select="Heure_Admission"/></li>
        <li>Date : <xsl:apply-templates select="Date_Admission"/></li>
        <li>Nom de l'institution : <xsl:apply-templates select="Nom_Institution_Admission"/></li>
        <li>Adresse de l'institution : <xsl:apply-templates select="Adresse_Institution_Admission"/></li>
        <li>Departement : <xsl:apply-templates select="Departement_Admission"/></li>
        <li>Description : <xsl:apply-templates select="Description_Admission"/></li>
      </ul>
    </ul>
  </xsl:template>

```

```

    </ul>
  </ul>
</xsl:template>

<xsl:template match="Medecin">
  <ul type="square">
    <li><p align="left"><u>Medecin Traitant</u></p></li>
    <ul type="disc">
      <li>Nom : <xsl:apply-templates select="Nom_Medecin"/></li>
      <li>Prenom : <xsl:apply-templates select="Prenom_Medecin"/></li>
      <li>Numero INAMI : <xsl:apply-templates select="Numero_INAMI_Medecin"/></li>
      <li>Adresse : <xsl:apply-templates select="Adresse_Medecin"/></li>
    </ul>
  </ul>
</xsl:template>

<xsl:template match="Examen">
  <li><a xsl:use-attribute-sets="href">
    <u><xsl:apply-templates select="Titre_EMM"/></u></a>
  </li>
</xsl:template>

<xsl:attribute-set name="href">
  <xsl:attribute name="href">Examen1.htm</xsl:attribute>
</xsl:attribute-set>
</xsl:stylesheet>

```

- La feuille de style « EMMCHhtm.xsl » est la suivante :

```

<?xml version="1.0" encoding="UTF-8"?>
<xsl:stylesheet version="1.0" xmlns:xsl="http://www.w3.org/1999/XSL/Transform">
  <xsl:template match="EMMCH">
    <html>
      <body>
        <h2 align="center">
          <font color="#0080FF"><xsl:apply-templates select="Titre_EMM"/></font></h2>
          <xsl:apply-templates select="Enregistrement"/>
          <xsl:apply-templates select="Contexte"/>
          <ul type="square">
            <li><p align="left"><u>Prescripteurs</u></p></li>
            <xsl:apply-templates select="Prescripteur"/>
          </ul>
          <blockquote><xsl:call-template name="CreateTable"/></blockquote>
          <hr></hr>
          <xsl:variable name="Value">
            <xsl:value-of select="./Diagnostic"/>
          </xsl:variable>
          <xsl:if test="not( $Value = " )">
            <ul type="square">
              <li><p align="left"><u>Diagnostic</u></p></li>
              <blockquote><xsl:value-of select="./Diagnostic"/></blockquote>
            </ul>
          </xsl:if>
          <xsl:variable name="Value"><xsl:value-of select="./Commentaire_General"/></xsl:variable>
          <xsl:if test="not( $Value = " )">
            <ul type="square">
              <li><p align="left"><u>Commentaire</u></p></li>
              <blockquote><xsl:value-of select="./Commentaire_General"/></blockquote>
            </ul>
          </xsl:if>

```

```

        <ul type="square"><li><p align="left"><u>Observations</u></p></li></ul>
        <xsl:apply-templates select="Observateur"/>
    </body>
</html>
</xsl:template>
<xsl:template match="Enregistrement">
    <blockquote>
        Reference : <xsl:apply-templates select="Numero_Enregistrement"/><br></br>
        Date : <xsl:apply-templates select="Date_Enregistrement"/><br></br>
        Heure : <xsl:apply-templates select="Heure_Enregistrement"/><br></br>
        Createur : <xsl:apply-templates select="Nom_Institution_Enregistrement"/>
            ( <xsl:apply-templates select="Adresse_Institution_Enregistrement"/> )<br></br>
    </blockquote>
    <hr></hr>
</xsl:template>

<xsl:template match="Contexte">
    <ul type="square">
        <li><p align="left"><u>Contexte</u></p></li>
        <ul type="disc">
            <li>Date de prelevement : <xsl:apply-templates select="Date_Prelevement"/></li>
            <li>Mode de prelevement : <xsl:apply-templates select="Mode_Prelevement"/></li>
            <li>Type de technique : <xsl:apply-templates select="Nature_Prelevement"/></li>
            <li>Organe : <xsl:apply-templates select="Site_Prelevement"/></li>
            <li>Contexte clinique : <xsl:apply-templates select="Contexte_Clinique"/></li>
            <li>Recherche particuliere : <xsl:apply-templates select="Recherche_Particuliere"/></li>
        </ul>
    </ul>
</xsl:template>

<xsl:template match="Prescripteur">
    <ul type="disc">
        <li>Nom : <xsl:apply-templates select="Nom_Prescripteur"/></li>
        <li>Prenom : <xsl:apply-templates select="Prenom_Prescripteur"/></li>
        <li>Adresse : <xsl:apply-templates select="Adresse_Prescripteur"/></li>
    </ul>
    <br></br>
</xsl:template>

<xsl:template name="CreateTable">
    <table>
        <xsl:call-template name="CreateNewRow">
            <xsl:with-param name="position">1</xsl:with-param>
            <xsl:with-param name="nbrCol">8</xsl:with-param>
        </xsl:call-template>
    </table>
</xsl:template>

<xsl:template name="CreateNewRow">
    <xsl:param name="position"/>
    <xsl:param name="nbrCol"/>
    <!-- If there are any Media with a position number of $position -->
    <xsl:if test="Media[position()=$position]">
    <tr>
        <xsl:call-template name="CreateRow">
            <xsl:with-param name="position"><xsl:value-of select="$position"/></xsl:with-param>
            <xsl:with-param name="nbrCol"><xsl:value-of select="$nbrCol"/></xsl:with-param>
            <xsl:with-param name="nbrColLib"><xsl:value-of select="$nbrCol"/></xsl:with-param>
        </xsl:call-template>
    </tr>

```

```

<xsl:call-template name="CreateNewRow">
  <xsl:with-param name="position"><xsl:value-of select="$position + $nbrCol"/></xsl:with-param>
  <xsl:with-param name="nbrCol"><xsl:value-of select="$nbrCol"/></xsl:with-param>
</xsl:call-template>
</xsl:if>
</xsl:template>

<xsl:template name="CreateRow">
  <xsl:param name="position"/>
  <xsl:param name="nbrCol"/>
  <xsl:param name="nbrColLib"/>
  <xsl:if test="$nbrColLib > 0">
    <xsl:variable name="Type">
      <xsl:value-of select="Media[position()=$position]/Type_Media"/>
    </xsl:variable>
    <xsl:choose>
      <xsl:when test="$Type = 'Image/Tif'">
        <xsl:apply-templates select="Media[position()=$position]"/>
        <xsl:call-template name="CreateRow">
          <xsl:with-param name="position"><xsl:value-of select="$position + 1"/></xsl:with-param>
          <xsl:with-param name="nbrCol"><xsl:value-of select="$nbrCol"/></xsl:with-param>
          <xsl:with-param name="nbrColLib"><xsl:value-of select="$nbrColLib - 1"/></xsl:with-param>
        </xsl:call-template>
      </xsl:when>
      <xsl:when test="$Type = 'Image/Jpg'">
        <xsl:apply-templates select="Media[position()=$position]"/>
        <xsl:call-template name="CreateRow">
          <xsl:with-param name="position"><xsl:value-of select="$position + 1"/></xsl:with-param>
          <xsl:with-param name="nbrCol"><xsl:value-of select="$nbrCol"/></xsl:with-param>
          <xsl:with-param name="nbrColLib"><xsl:value-of select="$nbrColLib - 1"/></xsl:with-param>
        </xsl:call-template>
      </xsl:when>
      <xsl:otherwise>
        <xsl:choose>
          <xsl:when test="Media[position()=$position]">
            <xsl:call-template name="CreateRow">
              <xsl:with-param name="position"><xsl:value-of select="$position + 1"/></xsl:with-param>
              <xsl:with-param name="nbrCol"><xsl:value-of select="$nbrCol"/></xsl:with-param>
              <xsl:with-param name="nbrColLib"><xsl:value-of select="$nbrColLib"/></xsl:with-param>
            </xsl:call-template>
          </xsl:when>
          <xsl:otherwise>
            <td></td>
            <xsl:call-template name="CreateRow">
              <xsl:with-param name="position"><xsl:value-of select="$position + 1"/></xsl:with-param>
              <xsl:with-param name="nbrCol"><xsl:value-of select="$nbrCol"/></xsl:with-param>
              <xsl:with-param name="nbrColLib"><xsl:value-of select="$nbrColLib - 1"/></xsl:with-param>
            </xsl:call-template>
          </xsl:otherwise>
        </xsl:choose>
      </xsl:otherwise>
    </xsl:choose>
  </xsl:if>
</xsl:template>

<xsl:template match="Media">
  <td>
    <a xsl:use-attribute-sets="href">
      <img xsl:use-attribute-sets="src" width="100" height="100" toolbar="off"/>
    </a>
  </td>
</xsl:template>

```

```

</td>
</xsl:template>

<xsl:attribute-set name="href">
  <xsl:attribute name="href">EM/<xsl:apply-templates select="Nom_Media"/></xsl:attribute>
</xsl:attribute-set>

<xsl:attribute-set name="src">
  <xsl:attribute name="src">EM/<xsl:apply-templates select="Nom_Media"/></xsl:attribute>
</xsl:attribute-set>

<xsl:template match="Nom_Media">
  <xsl:variable name="NomMediaTif"><xsl:value-of select="."/></xsl:variable>
  <xsl:variable name="NomMedia">
    <xsl:value-of select="substring-before($NomMediaTif,')"/>
  </xsl:variable>
  <xsl:variable name="NomMediaJpg"><xsl:value-of select="concat($NomMedia, '.jpg')"/></xsl:variable>
  <xsl:value-of select="$NomMediaJpg"/>
</xsl:template>

<xsl:template match="Observateur">
  <blockquote>
    <ul type="disc">
      <li>Nom : <xsl:apply-templates select="Nom_Observateur"/></li>
      <li>Prenom : <xsl:apply-templates select="Prenom_Observateur"/></li>
      <li>Adresse : <xsl:apply-templates select="Adresse_Observateur"/></li>
    </ul>
    <table border="2"> <xsl:apply-templates select="Observation"/> </table>
  </blockquote>
</xsl:template>

<xsl:template match="Observation">
  <tr>
    <td>
      <ul type="disc">
        <li>Description de la lesion : <xsl:apply-templates select="Description_Lesion"/></li>
        <li>Difficultes rencontrees : <xsl:apply-templates select="Difficultes_Rencontres"/></li>
        <li>Commentaire : <xsl:apply-templates select="Commentaire"/></li>
      </ul>
    </td>
    <td>
      <a xsl:use-attribute-sets="href2">
        <img xsl:use-attribute-sets="src2" width="200" height="200" toolbar="off"/></a>
      </td>
  </tr>
</xsl:template>

<xsl:attribute-set name="href2">
  <xsl:attribute name="href">EM/<xsl:apply-templates select="Media/Nom_Media"/></xsl:attribute>
</xsl:attribute-set>

<xsl:attribute-set name="src2">
  <xsl:attribute name="src">EM/<xsl:apply-templates select="Media/Nom_Media"/></xsl:attribute>
</xsl:attribute-set>
</xsl:stylesheet>

```

- La feuille de style « FichierIni.xsl » est la suivante :

```

<?xml version="1.0" encoding="UTF-8"?>
<xsl:stylesheet version="1.0" xmlns:xsl="http://www.w3.org/1999/XSL/Transform">

```

```

<xsl:output method = "html"/>
<xsl:variable name="numero" select="'90041'"/>
<xsl:template match="EMMCH">
  [Commun]
  Date = <xsl:apply-templates select="Enregistrement/Date_Enregistrement"/>
  Heure = <xsl:apply-templates select="Enregistrement/Heure_Enregistrement"/>
  Titre = <xsl:apply-templates select="Titre_EMM"/>
  Comment = <ul type="square">
    <li><p align="left"><u>Diagnostic</u></p></li>
    <blockquote> <xsl:value-of select="./Diagnostic"/> </blockquote>
  </ul>
  <ul type="square">
    <li><p align="left"><u>Commentaire</u></p></li>
    <blockquote> <xsl:value-of select="./Commentaire_General"/> </blockquote>
  </ul>
  Site_Emetteur = <xsl:value-of select="$numero"/>
  Liste = <xsl:call-template name="CreateList"/>
    <xsl:apply-templates select="Media"/>
</xsl:template>

<xsl:template name="CreateList">
  <xsl:variable name="position" select="1"/>
  <xsl:for-each select="Media"><xsl:value-of select="$numero"/><xsl:value-of select="$position"/>;
  <xsl:variable name="position" select="$position + 1"/>
  </xsl:for-each>
</xsl:template>

<xsl:template match="Media">
  <xsl:variable name="nom"><xsl:value-of select="Nom_Media"/></xsl:variable>
  [<xsl:value-of select="$numero"/><xsl:value-of select="position()"/>]
  Createur = <xsl:apply-templates select="../Enregistrement/Nom_Institution_Enregistrement"/>
  Date_Modif = <xsl:apply-templates select="Date_Acquisition"/>
  Date_User = <xsl:apply-templates select="Date_Acquisition"/>
  Diapo_Lame = L
  Reference = <xsl:apply-templates select="../Enregistrement/Numero_Enregistrement"/>
  Responsable = <xsl:apply-templates select="../Prescripteur/Nom_Prescripteur"/>
    <xsl:apply-templates select="../Prescripteur/Prenom_Prescripteur"/>
  Document_Type = 16
  Prelevement = <xsl:apply-templates select="../Contexte/Mode_Prelevement"/>
  Technique = <xsl:apply-templates select="../Contexte/Nature_Prelevement"/>
  Organe = <xsl:apply-templates select="../Contexte/Site_Prelevement"/>
  Lesion1 = NULL
  Lesion2 = NULL
  Lesion3 = NULL
  Legende = <xsl:apply-templates select="..*/Observation[ Media/Nom_Media = $nom
    ]/Description_Lesion"/>
  Comment = <xsl:apply-templates select="..*/Observation[ Media/Nom_Media = $nom ]/Commentaire"/>
  Source_Image = <xsl:apply-templates select="Emplacement_Media"/>
  Source_Icône = <xsl:apply-templates select="Emplacement_Media"/>
</xsl:template>

<xsl:template match="Emplacement_Media">
  <xsl:variable name="EmplacementMediaTif"><xsl:value-of select="."/></xsl:variable>
  <xsl:variable name="EmplacementMedia">
    <xsl:value-of select="substring-before($EmplacementMediaTif,')"/>
  </xsl:variable>
  <xsl:variable name="EmplacementMediaJpg">
    <xsl:value-of select="concat($EmplacementMedia, '.jpg')"/>
  </xsl:variable>
  <xsl:value-of select="$EmplacementMediaJpg"/>

```

```
</xsl:template>
</xsl:stylesheet>
```

Scénario réservé à la transmission via e-mail

- Le fichier XML de coordination de ce scénario est le suivant :

```
<Scenario>
  <Acces>htm2</Acces>
  <Emplacement_Presentation>Scenario.xml</Emplacement_Presentation>
  <Transformation>
    <Input type="primaire" href="DMM.xml">DMM.xml</Input>
    <Input type="secondaire" href="EMMCH.xml">EMMCH.xml</Input>
    <Feuille type="xsl" href="DMMEMMCHhtm.xml">DMMEMMCHhtm.xml</Feuille>
    <Output type="primaire" href="DMMEexamen1.htm">DMMEexamen1.htm</Output>
  </Transformation>
</Scenario>
```

On remarque que ce scénario a comme identificateur (attribut « Acces ») le valeur « htm2 ». Ce scénario est composé d'une seule transformation de type XSLT car l'attribut « type » de l'élément « Feuille » de cette transformation possède la valeur « xsl ».

Cette transformation utilise la feuille de style « DMMEMMCHhtm.xml » pour transformer les fichiers sources « DMM.xml » et « EMMCH.xml » qui respectent les DTD correspondant au niveau 1 et 2, c'est-à-dire regroupant les informations associées à tout le document multimédia et à l'examen en cytologie hématologique. On obtient alors un fichier cible nommé « DMMEexamen1.htm ».

- La feuille de style « DMMEMMCHhtm.xml » est la suivante :

```
<?xml version="1.0" encoding="UTF-8"?>
<xsl:stylesheet version="1.0" xmlns:xsl="http://www.w3.org/1999/XSL/Transform">
  <xsl:template match="DMM">
    <html>
      <body>
        <h1 align="center"><xsl:apply-templates select="Origine_DMM"/></h1>
        <h2 align="center"><font color="#0080FF"><xsl:apply-templates select="Titre_DMM"/></font></h2>
        <h3 align="center"><font color="#0080FF">
          <xsl:apply-templates select="Date_Creation_DMM"/></font></h3>
        <hr></hr>
        <h4 align="left"><u>Informations Generales</u></h4>
        <xsl:apply-templates select="Patient"/>
        <hr></hr>
        <h4><u>L'examen</u></h4>
        <ul type="square">
          <xsl:apply-templates select="document('../EMMCH.xml')"/>
        </ul>
      </body>
    </html>
  </xsl:template>

  <xsl:template match="Patient">
    <ul type="square">
      <li><p align="left"><u>Patient</u></p></li>
      <ul type="disc">
        <li>Numero : <xsl:apply-templates select="Numero_Patient"/></li>
        <li>Nom : <xsl:apply-templates select="Nom_Patient"/></li>
        <li>Prenom : <xsl:apply-templates select="Prenom_Patient"/></li>
      </ul>
    </ul>
  </xsl:template>
```

```

<li>Sexe : <xsl:apply-templates select="Sexe_Patient"/></li>
<li>Nom de jeune fille :<xsl:apply-templates select="Nom_Jeune_Fille_Patient"/></li>
<li>Taille : <xsl:apply-templates select="Taille_Patient"/></li>
<li>Poids : <xsl:apply-templates select="Poids_Patient"/></li>
<li>Date de naissance : <xsl:apply-templates select="Date_Naissance_Patient"/></li>
<li>Adresse : <xsl:apply-templates select="Adresse_Patient"/></li>
<li>Telephone1 : <xsl:apply-templates select="Telephone1_Patient"/></li>
<li>Telephone2 : <xsl:apply-templates select="Telephone2_Patient"/></li>
<li>Mutuel : <xsl:apply-templates select="Mutuel_Patient"/></li>
<li>Coordonnees de facturation :
  <xsl:apply-templates select="Coordonnees_Facturation_Patient"/>
</li>
<li>Commentaire : <xsl:apply-templates select="Commentaire_Patient"/></li>
</ul>
</ul>
<xsl:apply-templates select="Admission"/>
<xsl:apply-templates select="Medecin"/>
</xsl:template>

<xsl:template match="Admission">
  <ul type="square">
    <li><p align="left"><u>Admission</u></p></li>
    <ul type="disc">
      <li>Heure : <xsl:apply-templates select="Heure_Admission"/></li>
      <li>Date : <xsl:apply-templates select="Date_Admission"/></li>
      <li>Nom de l'institution : <xsl:apply-templates select="Nom_Institution_Admission"/></li>
      <li>Adresse de l'institution : <xsl:apply-templates select="Adresse_Institution_Admission"/></li>
      <li>Departement : <xsl:apply-templates select="Departement_Admission"/></li>
      <li>Description : <xsl:apply-templates select="Description_Admission"/></li>
    </ul>
  </ul>
</xsl:template>

<xsl:template match="Medecin">
  <ul type="square">
    <li><p align="left"><u>Medecin Traitant</u></p></li>
    <ul type="disc">
      <li>Nom : <xsl:apply-templates select="Nom_Medecin"/></li>
      <li>Prenom : <xsl:apply-templates select="Prenom_Medecin"/></li>
      <li>Numero INAMI : <xsl:apply-templates select="Numero_Medecin"/></li>
      <li>Adresse : <xsl:apply-templates select="Adresse_Medecin"/></li>
    </ul>
  </ul>
</xsl:template>

<xsl:template match="EMMCH">
  <li><font color="#0080FF"><xsl:apply-templates select="Titre_EMM"/></font></li>
  <xsl:apply-templates select="Enregistrement"/>
  <xsl:apply-templates select="Contexte"/>
  <blockquote>
    <p align="left"><u>Prescripteurs</u></p>
    <xsl:apply-templates select="Prescripteur"/>
  </blockquote>
  <xsl:variable name="Value"><xsl:value-of select="./Diagnostic"/></xsl:variable>
  <xsl:if test="not( $Value = " )">
    <blockquote>
      <p align="left"><u>Diagnostic</u></p>
      <ul type="disc">
        <li><xsl:value-of select="./Diagnostic"/></li>
      </ul>
    </blockquote>
  </xsl:if>

```



```

    </blockquote>
</xsl:if>
<xsl:variable name="Value"><xsl:value-of select="./Commentaire_General"/></xsl:variable>
<xsl:if test="not( $Value = " )">
  <blockquote>
    <p align="left"><u>Commentaire</u></p>
    <ul type="disc">
      <li><xsl:value-of select="./Commentaire_General"/></li>
    </ul>
  </blockquote>
</xsl:if>
<blockquote><p align="left"><u>Observations</u></p></blockquote>
<xsl:apply-templates select="Observateur"/>
</xsl:template>

<xsl:template match="Enregistrement">
  <blockquote>
    Reference : <xsl:apply-templates select="Numero_Enregistrement"/><br></br>
    Date : <xsl:apply-templates select="Date_Enregistrement"/><br></br>
    Heure : <xsl:apply-templates select="Heure_Enregistrement"/><br></br>
    Createur : <xsl:apply-templates select="Nom_Institution_Enregistrement"/>
      ( <xsl:apply-templates select="Adresse_Institution_Enregistrement"/> )<br></br>
  </blockquote>
</xsl:template>

<xsl:template match="Contexte">
  <blockquote>
    <p align="left"><u>Contexte</u></p>
    <ul type="disc">
      <li>Date de prelevement : <xsl:apply-templates select="Date_Prelevement"/></li>
      <li>Mode de prelevement : <xsl:apply-templates select="Mode_Prelevement"/></li>
      <li>Type de technique : <xsl:apply-templates select="Nature_Prelevement"/></li>
      <li>Organe : <xsl:apply-templates select="Site_Prelevement"/></li>
      <li>Contexte clinique : <xsl:apply-templates select="Contexte_Clinique"/></li>
      <li>Recherche particuliere : <xsl:apply-templates select="Recherche_Particuliere"/></li>
    </ul>
  </blockquote>
</xsl:template>

<xsl:template match="Prescripteur">
  <ul type="disc">
    <li>Nom : <xsl:apply-templates select="Nom_Prescripteur"/></li>
    <li>Prenom : <xsl:apply-templates select="Prenom_Prescripteur"/></li>
    <li>Adresse : <xsl:apply-templates select="Adresse_Prescripteur"/></li>
  </ul>
</xsl:template>

<xsl:template match="Observateur">
  <blockquote>
    <ul type="disc">
      <li>Nom : <xsl:apply-templates select="Nom_Observateur"/></li>
      <li>Prenom : <xsl:apply-templates select="Prenom_Observateur"/></li>
      <li>Adresse : <xsl:apply-templates select="Adresse_Observateur"/></li>
    </ul>
    <br></br>
    <table border="2">
      <xsl:apply-templates select="Observation"/>
    </table>
    <hr></hr>
  </blockquote>

```

```

</xsl:template>

<xsl:template match="Observation">
  <tr>
    <td>
      <ul type="disc">
        <li>Description de la lesion : <xsl:apply-templates select="Description_Lesion"/></li>
        <li>Difficultes rencontrees : <xsl:apply-templates select="Difficultes_Rencontrees"/></li>
        <li>Commentaire : <xsl:apply-templates select="Commentaire"/></li>
      </ul>
    </td>
    <td>
      <a xsl:use-attribute-sets="href">EM/<xsl:apply-templates select="Media/Nom_Media"/></a>
    </td>
  </tr>
</xsl:template>

<xsl:attribute-set name="href">
  <xsl:attribute name="href">EM/<xsl:apply-templates select="Media/Nom_Media"/></xsl:attribute>
</xsl:attribute-set>
</xsl:stylesheet>

```

Scénario réservé à la transmission vers « Télémis »

- Le fichier XML de coordination de ce scénario est le suivant :

```

<Scenario>
  <Acces>htm3</Acces>
  <Emplacement_Presentation>Scenario.xml</Emplacement_Presentation>
  <Transformation>
    <Input type="primaire" href="DMM.xml">DMM.xml</Input>
    <Input type="secondaire" href="EMMCH.xml">EMMCH.xml</Input>
    <Feuille type="xsl" href="DMMEMMCHlighthtm.xml">DMMEMMCHlighthtm.xml</Feuille>
    <Output type="primaire" href="DMMExamen1light.htm">DMMExamen1light.htm</Output>
  </Transformation>
</Scenario>

```

On remarque que ce scénario a comme identificateur (attribut « Acces ») le valeur « htm3 ». Ce scénario est composé d'une seule transformation de type XSLT car l'attribut « type » de l'élément « Feuille » de cette transformation possède la valeur « xsl ».

Cette transformation utilise la feuille de style « DMMEMMCHlighthtm.xml » pour transformer les fichiers sources « DMM.xml » et « EMMCH.xml » qui respectent les DTD correspondant au niveau 1 et 2, c'est-à-dire regroupant les informations associées à tout le document multimédia et à l'examen en cytologie hématologique. On obtient alors un fichier cible nommé « DMMExamen1light.htm ».

Cependant, ce fichier de coordination est incomplet car, durant la composition d'un document, lorsque le scénario est associé à un examen en cytologie hématologique, une transformation est ajoutée pour chaque image. Ces transformations utilisent la feuille de style « *Nom_Image.xml* » pour transformer le fichier « EMMCH.xml ». On obtient alors des fichiers « *Nom_Image.htm* » :

```

<Transformation>
  <Input type="primaire" href="EMMCH.xml">EMMCH.xml</Input>
  <Feuille type="xsl" href="Nom_Image.xml">Nom_Image.xml</Feuille>
  <Output type="secondaire" href="Nom_Image.htm">Nom_Image.htm</Output>
</Transformation>

```

(où *Nom_Image* doit être remplacé par le nom du fichier contenant l'image, sans son extension).

- La feuille de style « DMMEMMCHlighthtm.xsl » est la suivante :

```
<?xml version="1.0" encoding="UTF-8"?>
<xsl:stylesheet version="1.0" xmlns:xsl="http://www.w3.org/1999/XSL/Transform">
  <xsl:template match="DMM">
    <html>
      <body>
        <h1 align="center"><xsl:apply-templates select="Origine_DMM"/></h1>
        <h2 align="center">
          <font color="#0080FF"><xsl:apply-templates select="Titre_DMM"/></font></h2>
        <h3 align="center">
          <font color="#0080FF"><xsl:apply-templates select="Date_Creation_DMM"/></font></h3>
        <hr></hr>
        <h4 align="left"><u>Informations Generales</u></h4>
        <xsl:apply-templates select="Patient"/>
        <hr></hr>
        <h4><u>L'examen</u></h4>
        <ul type="square">
          <xsl:apply-templates select="document('../EMMCH.xml')"/>
        </ul>
      </body>
    </html>
  </xsl:template>

  <xsl:template match="Patient">
    <ul type="square">
      <li><p align="left"><u>Patient</u></p></li>
      <ul type="disc">
        <li>Numero : <xsl:apply-templates select="Numero_Patient"/></li>
        <li>Nom : <xsl:apply-templates select="Nom_Patient"/></li>
        <li>Prenom : <xsl:apply-templates select="Prenom_Patient"/></li>
        <li>Sexe : <xsl:apply-templates select="Sexe_Patient"/></li>
        <li>Nom de jeune fille : <xsl:apply-templates select="Nom_Jeune_Fille_Patient"/></li>
        <li>Taille : <xsl:apply-templates select="Taille_Patient"/></li>
        <li>Poids : <xsl:apply-templates select="Poids_Patient"/></li>
        <li>Date de naissance : <xsl:apply-templates select="Date_Naissance_Patient"/></li>
        <li>Adresse : <xsl:apply-templates select="Adresse_Patient"/></li>
        <li>Telephone1 : <xsl:apply-templates select="Telephone1_Patient"/></li>
        <li>Telephone2 : <xsl:apply-templates select="Telephone2_Patient"/></li>
        <li>Mutuel : <xsl:apply-templates select="Mutuel_Patient"/></li>
        <li>Coordonnees de facturation :
          <xsl:apply-templates select="Coordonnees_Facturation_Patient"/></li>
        <li>Commentaire : <xsl:apply-templates select="Commentaire_Patient"/></li>
      </ul>
    </ul>
    <xsl:apply-templates select="Admission"/>
    <xsl:apply-templates select="Medecin"/>
  </xsl:template>

  <xsl:template match="Admission">
    <ul type="square">
      <li><p align="left"><u>Admission</u></p></li>
      <ul type="disc">
        <li>Heure : <xsl:apply-templates select="Heure_Admission"/></li>
        <li>Date : <xsl:apply-templates select="Date_Admission"/></li>
      </ul>
    </ul>
  </xsl:template>
```

```

        <li>Nom de l'institution : <xsl:apply-templates select="Nom_Institution_Admission"/></li>
        <li>Adresse de l'institution : <xsl:apply-templates select="Adresse_Institution_Admission"/></li>
        <li>Departement : <xsl:apply-templates select="Departement_Admission"/></li>
        <li>Description : <xsl:apply-templates select="Description_Admission"/></li>
    </ul>
</ul>
</xsl:template>

<xsl:template match="Medecin">
    <ul type="square">
        <li><p align="left"><u>Medecin Traitant</u></p></li>
        <ul type="disc">
            <li>Nom : <xsl:apply-templates select="Nom_Medecin"/></li>
            <li>Prenom : <xsl:apply-templates select="Prenom_Medecin"/></li>
            <li>Numero INAMI : <xsl:apply-templates select="Numero_Medecin"/></li>
            <li>Adresse : <xsl:apply-templates select="Adresse_Medecin"/></li>
        </ul>
    </ul>
</xsl:template>

<xsl:template match="EMMCH">
    <li><font color="#0080FF"><xsl:apply-templates select="Titre_EMM"/></font></li>
    <xsl:apply-templates select="Enregistrement"/>
    <xsl:apply-templates select="Contexte"/>
    <blockquote>
        <p align="left"><u>Prescripteurs</u></p>
        <xsl:apply-templates select="Prescripteur"/>
    </blockquote>
    <xsl:variable name="Value"><xsl:value-of select="./Diagnostic"/></xsl:variable>
    <xsl:if test="not( $Value = " )">
        <blockquote>
            <p align="left"><u>Diagnostic</u></p>
            <ul type="disc">
                <li><xsl:value-of select="./Diagnostic"/></li>
            </ul>
        </blockquote>
    </xsl:if>
    <xsl:variable name="Value"><xsl:value-of select="./Commentaire_General"/></xsl:variable>
    <xsl:if test="not( $Value = " )">
        <blockquote>
            <p align="left"><u>Commentaire</u></p>
            <ul type="disc">
                <li><xsl:value-of select="./Commentaire_General"/></li>
            </ul>
        </blockquote>
    </xsl:if>
</xsl:template>

<xsl:template match="Enregistrement">
    <blockquote>
        Reference : <xsl:apply-templates select="Numero_Enregistrement"/><br></br>
        Date : <xsl:apply-templates select="Date_Enregistrement"/><br></br>
        Heure : <xsl:apply-templates select="Heure_Enregistrement"/><br></br>
        Createur : <xsl:apply-templates select="Nom_Institution_Enregistrement"/>
        ( <xsl:apply-templates select="Adresse_Institution_Enregistrement"/> )<br></br>
    </blockquote>
</xsl:template>

<xsl:template match="Contexte">
    <blockquote>

```

```

<p align="left"><u>Contexte</u></p>
<ul type="disc">
  <li>Date de prelevement : <xsl:apply-templates select="Date_Prelevement"/></li>
  <li>Mode de prelevement : <xsl:apply-templates select="Mode_Prelevement"/></li>
  <li>Type de technique : <xsl:apply-templates select="Nature_Prelevement"/></li>
  <li>Organe : <xsl:apply-templates select="Site_Prelevement"/></li>
  <li>Contexte clinique : <xsl:apply-templates select="Contexte_Clinique"/></li>
  <li>Recherche particuliere : <xsl:apply-templates select="Recherche_Particuliere"/></li>
</ul>
</blockquote>
</xsl:template>

<xsl:template match="Prescripteur">
  <ul type="disc">
    <li>Nom : <xsl:apply-templates select="Nom_Prescripteur"/></li>
    <li>Prenom : <xsl:apply-templates select="Prenom_Prescripteur"/></li>
    <li>Adresse : <xsl:apply-templates select="Adresse_Prescripteur"/></li>
  </ul>
</xsl:template>
</xsl:stylesheet>

```

- La feuille de style « *Nom_Image.xsl* » est la suivante :

```

<?xml version="1.0" encoding="UTF-8"?>
<xsl:stylesheet version="1.0" xmlns:xsl="http://www.w3.org/1999/XSL/Transform">
  <xsl:template match="EMMCH">
    <html>
      <body>
        <xsl:apply-templates select="Observateur"/>
      </body>
    </html>
  </xsl:template>

  <xsl:template match="Observateur">
    <xsl:apply-templates select="Observation[ Media/Nom_Media = 'Z' ]"/>
  </xsl:template>

  <xsl:template match="Observation">
    <blockquote>
      <ul type="disc">
        <li>Nom : <xsl:apply-templates select="../Nom_Observateur"/></li>
        <li>Prenom : <xsl:apply-templates select="../Prenom_Observateur"/></li>
        <li>Adresse : <xsl:apply-templates select="../Adresse_Observateur"/></li>
      </ul>
      <br><br>
      <table border="2">
        <tr>
          <td>
            <ul type="disc">
              <li>Description de la lesion : <xsl:apply-templates select="Description_Lesion"/></li>
              <li>Difficultes rencontrees : <xsl:apply-templates select="Difficultes_Rencontrees"/></li>
              <li>Commentaire : <xsl:apply-templates select="Commentaire"/></li>
            </ul>
          </td>
        </tr>
      </table>
      <hr></hr>
    </blockquote>
  </xsl:template>
</xsl:stylesheet>

```

(où **Z** est remplacé par le nom du fichier contenant l'image, avec son extension)

Annexe 11 : La composition d'un document via « Analysis »

The screenshot illustrates the workflow for creating a document in the 'analysis' software. The main window shows a grid of image buffers (labeled lama2, lama3, lama4, and Image 4) on the left. A red circle highlights the 'Composition' icon in the top toolbar. This icon leads to the 'Composition d'un document' dialog box, which asks if the user wants to create a new document containing the image buffers. The 'OK' button in this dialog leads to the 'Sélectionner répertoire' dialog box, where the user can choose a directory (currently 'Service'). The 'OK' button in this dialog leads to the 'Créer une page' dialog box, which contains various fields for document and patient information.

Composition d'un document

Voulez-vous créer un nouveau document contenant les images des buffers ?

OK Annuler

Chemin du nouveau document : C:\AnalyseAuto\Documents\Nouveau0 Parcourir

Sélectionner répertoire

Look in: Service

Répertoire: C:\Program Files\analysis\Module\Service

OK Cancel

Créer une page

Informations générales Informations de la page

Informations du document

Type : Volume :
 Titre : Version :
 Nom : Status :
 Origine : Date :

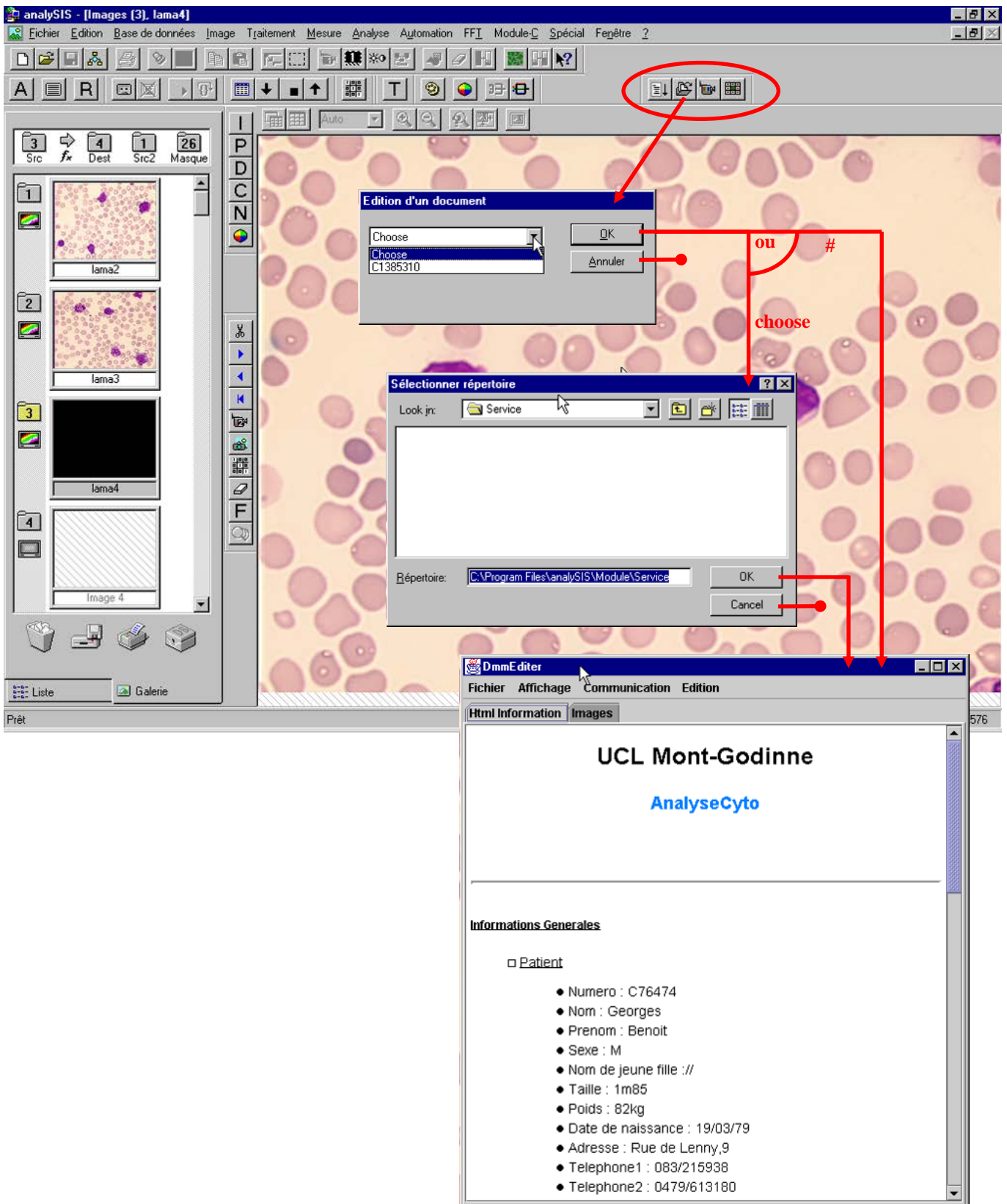
Informations du patient

Nomero : Medecin Traitant
 Nom : Numero :
 Prenom : Nom :
 Sexe : Adresse :
 Nom Jeune Fille :
 Taille :
 Poids :
 Date Naissance : Admission
 Adresse : Heure :
 Telephone 1 : Date :
 Telephone 2 : Nom de l'institut...
 Mutuel : Adresse de l'ins...
 Facturation : Departement :
 Commentaire : Description :

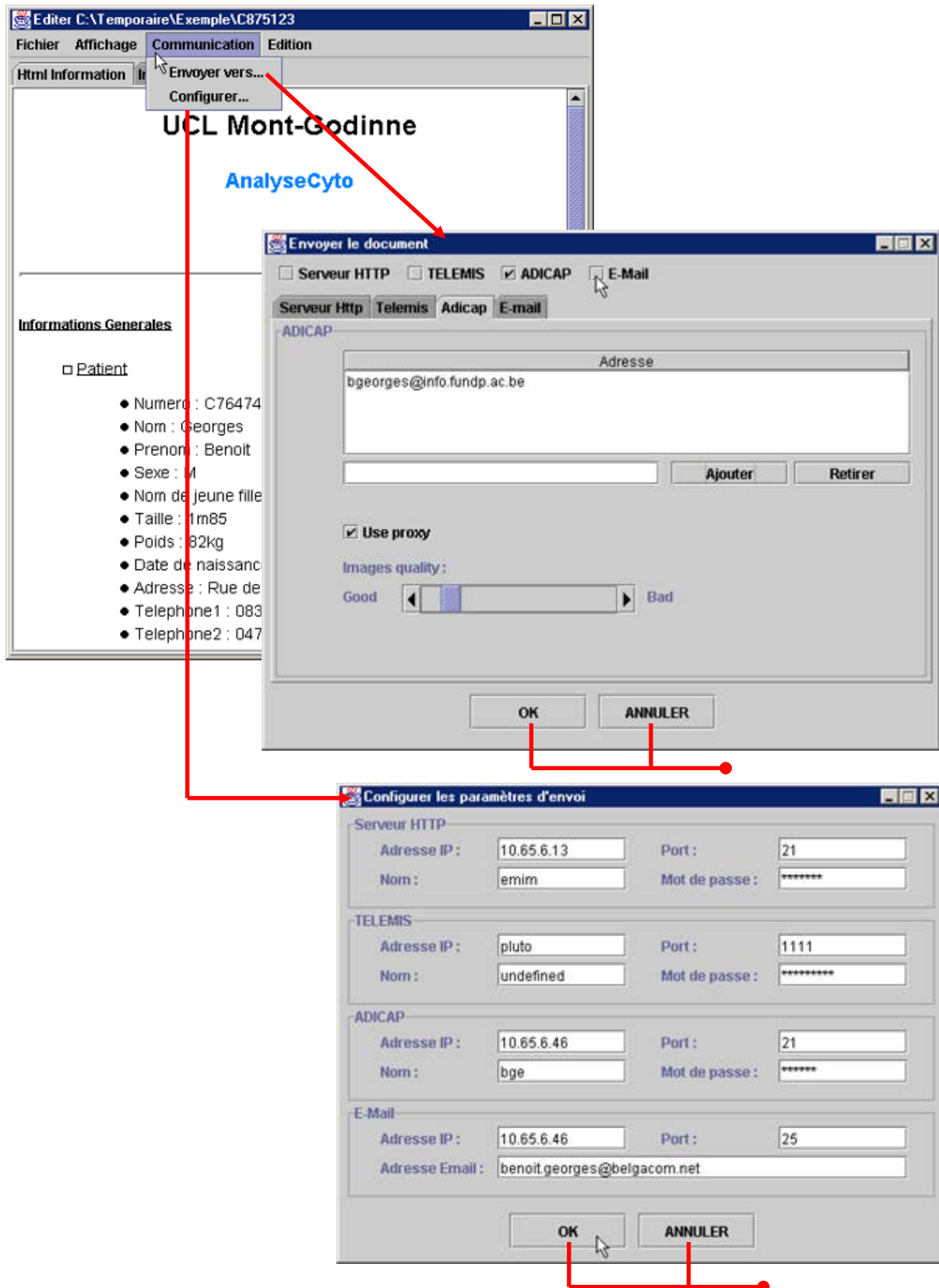
OK ANNULER

Annexe 12 : La visualisation, la transmission et l'édition d'un document via « Analysis »

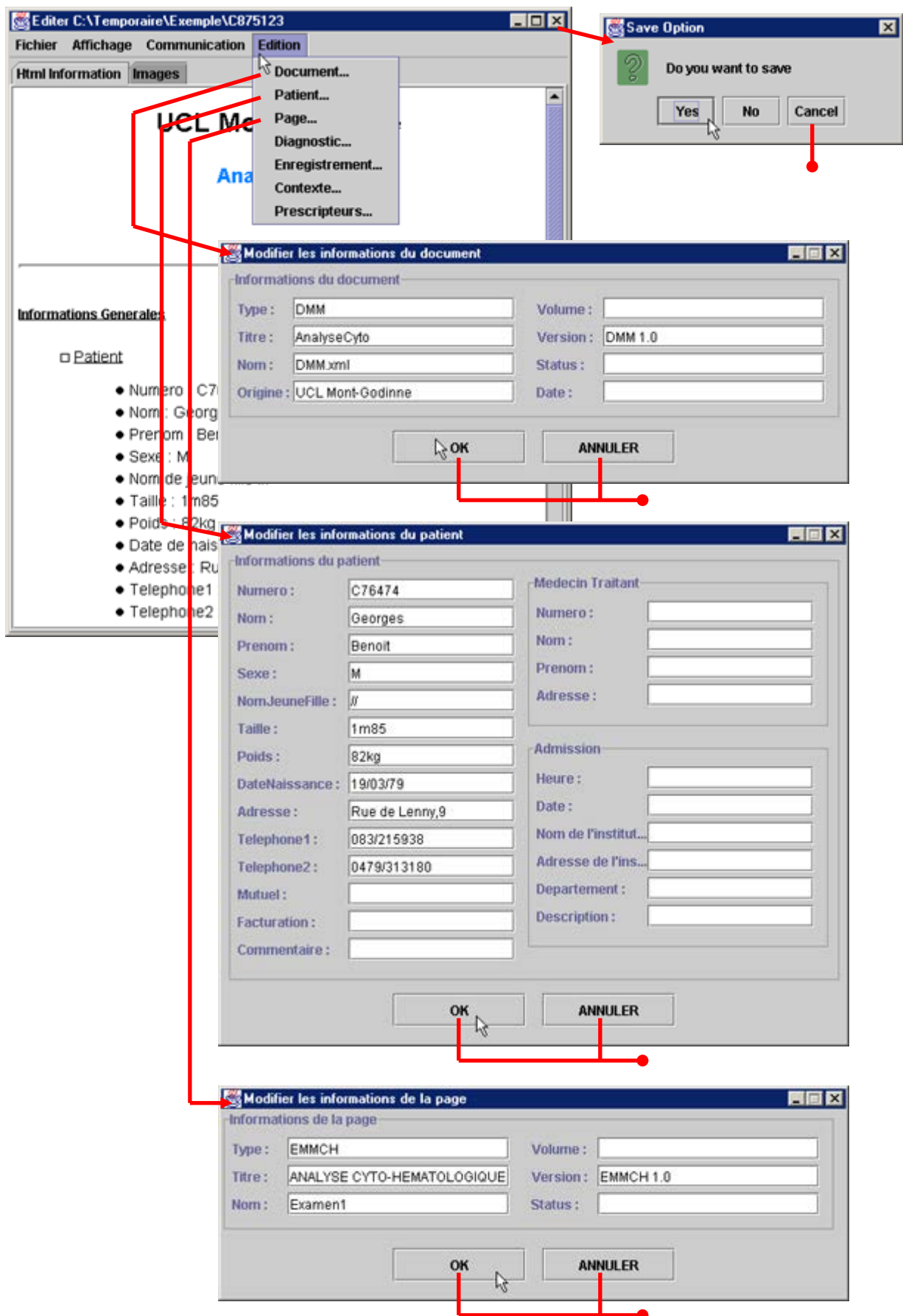
Visualisation d'un document multimédia médical

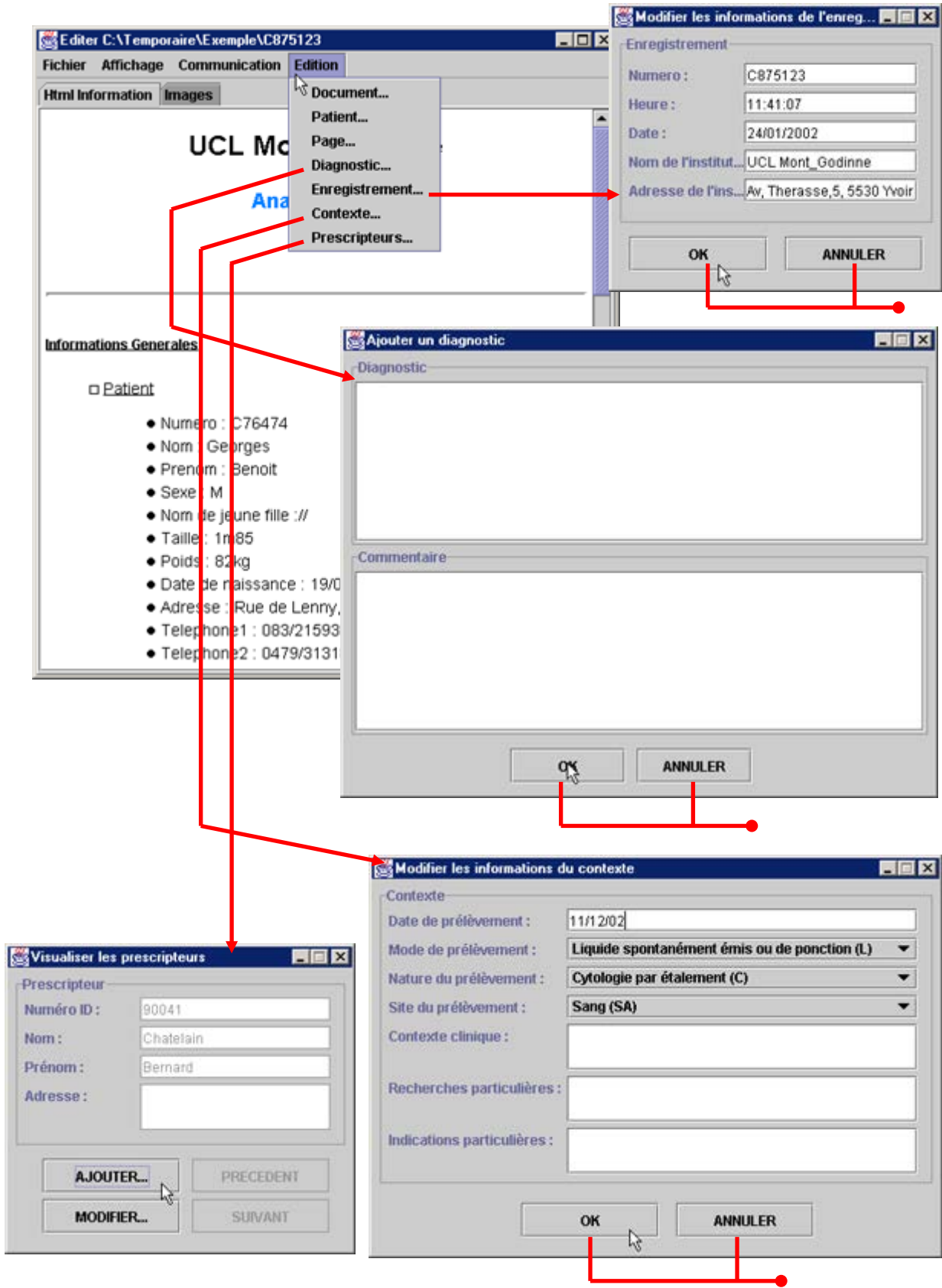


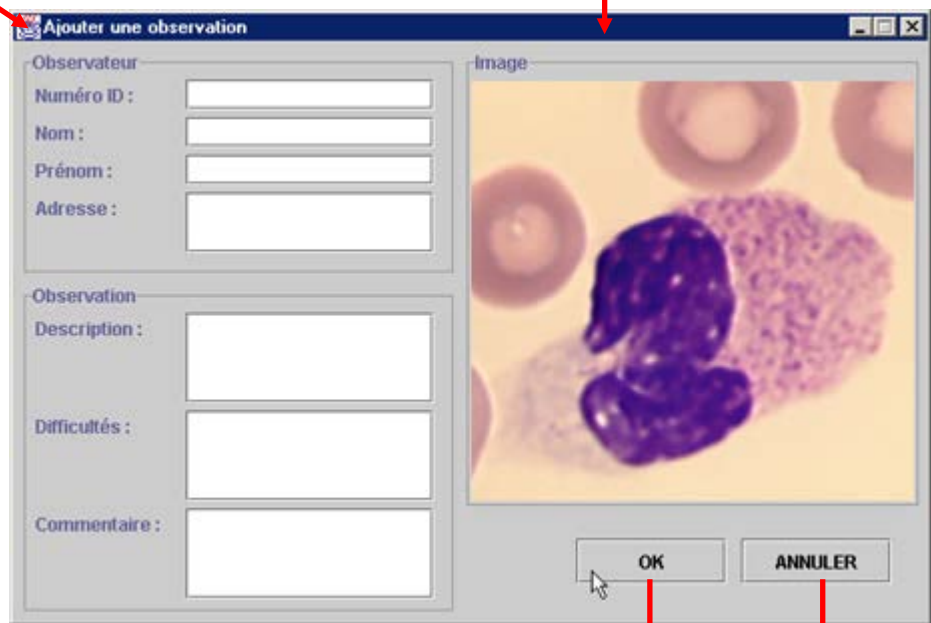
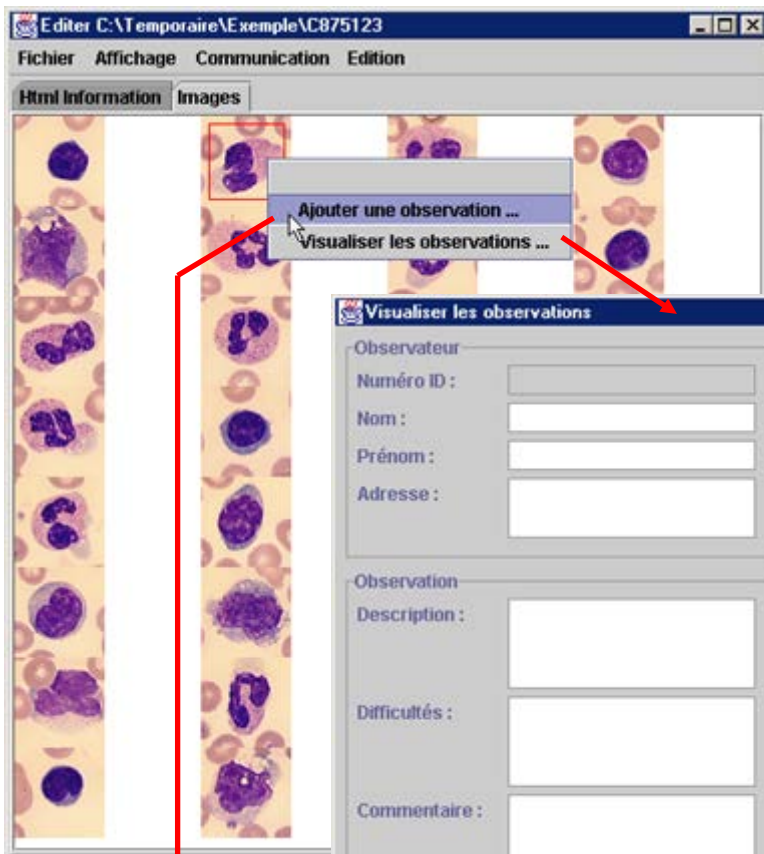
Transmission d'un document multimédia médical



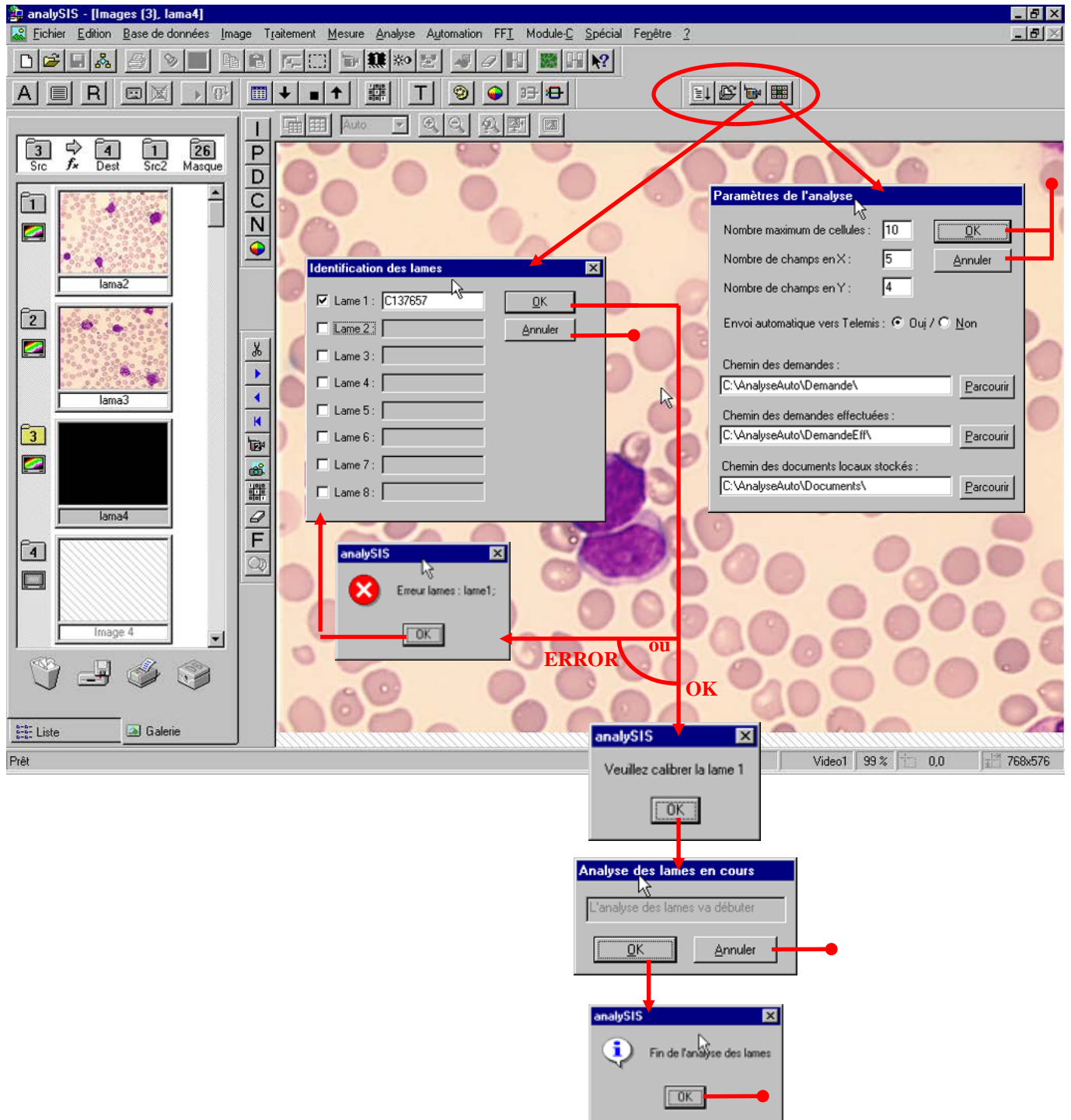
Edition d'un document multimédia médical



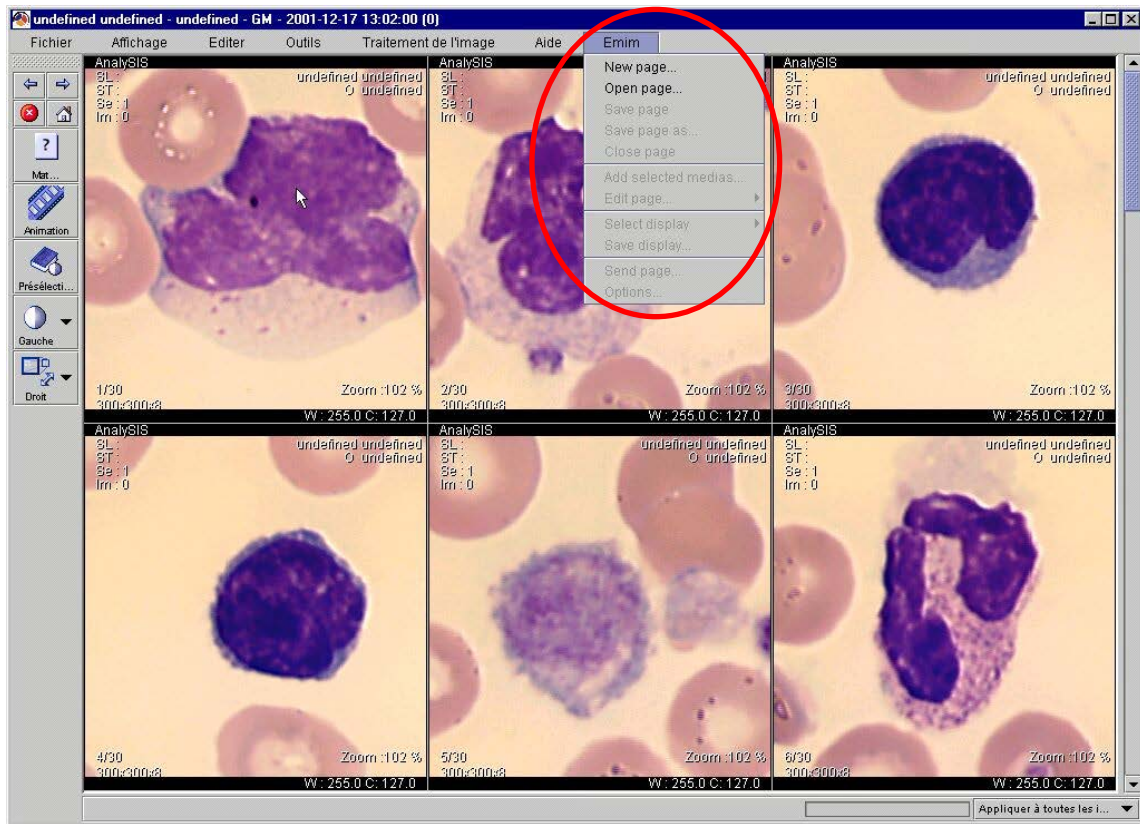




Annexe 13 : L'analyse automatique de lames via « Analysis »

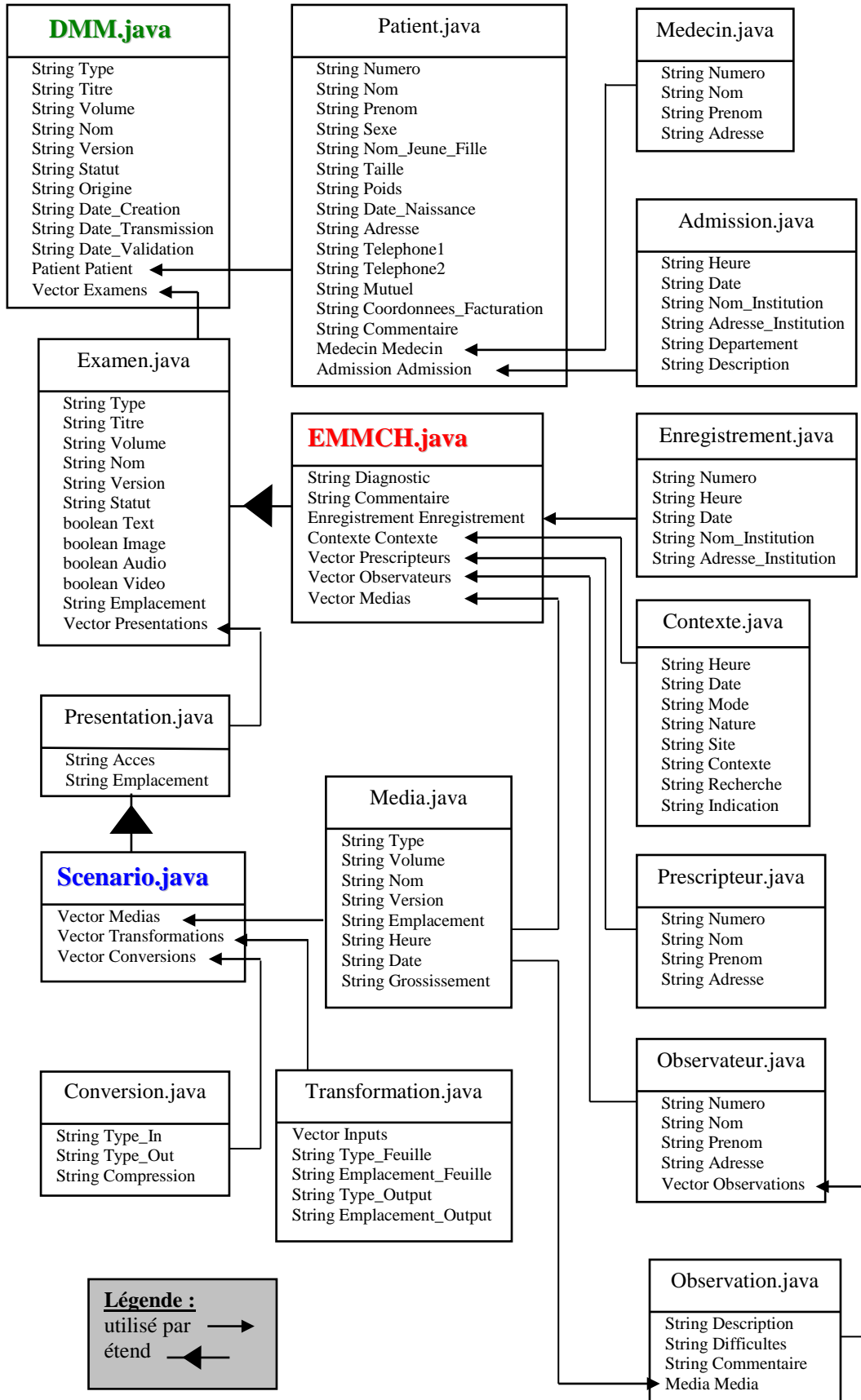


Annexe 14 : La composition, la visualisation, l'édition et la transmission d'un document via « Télémis »



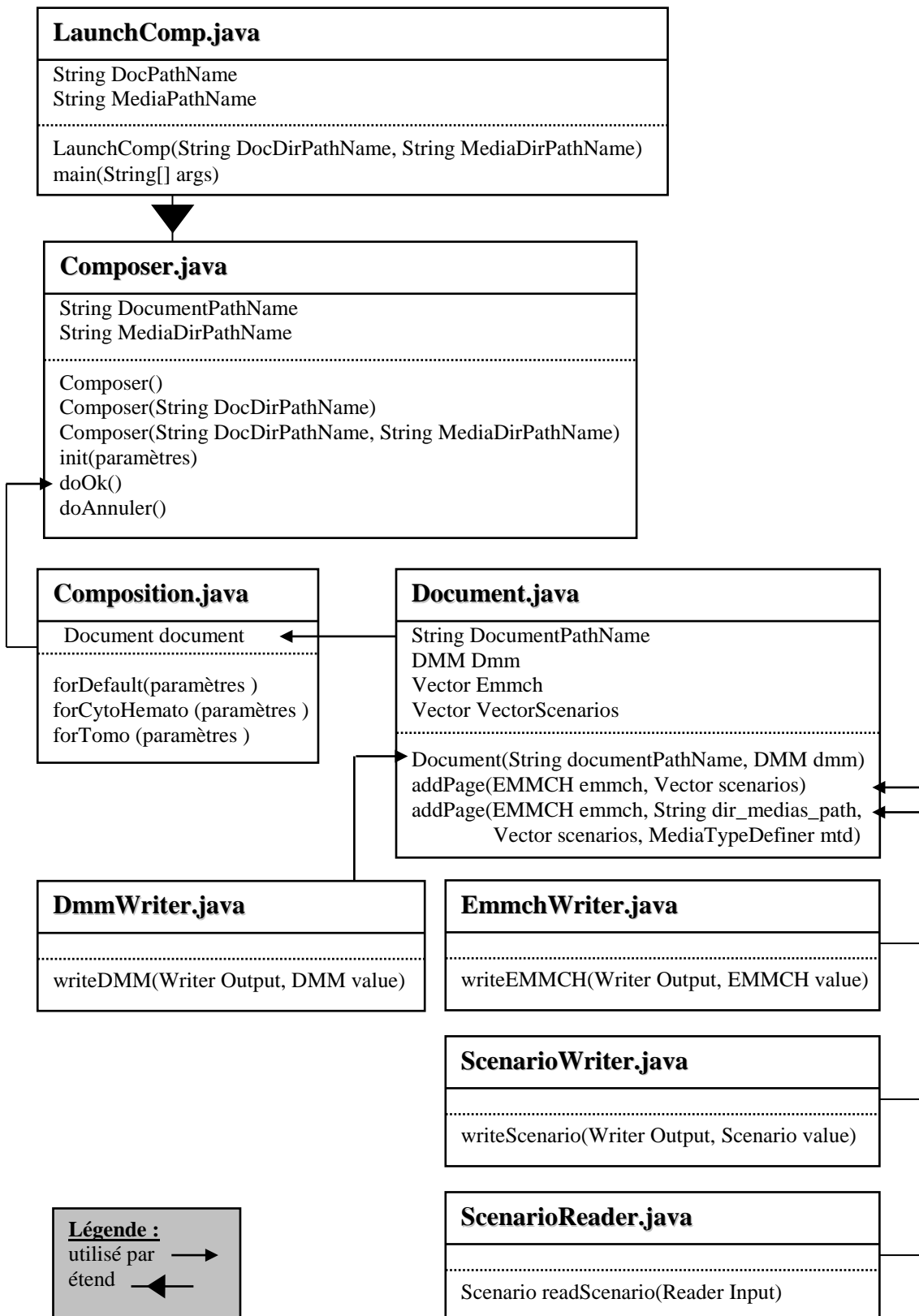
Annexe 15 : La structure du modèle de document multimédia médical en JAVA

Dans une application JAVA, cette structure peut être utilisée pour pouvoir mémoriser un document multimédia médical en mémoire vive. On peut remarquer que chaque entité du schéma E-A du modèle de document multimédia médical correspond à une "class" dans cette structure.



Annexe 16 : La composition d'un document multimédia médical en JAVA

Principales classes utilisées



1^{ère} Etape : Composition d'un document sans page

```
class Composition
{ public void forDefault( paramètres )
  { //Composition d'un objet DMM en mémoire
    DMM dmm = new DMM();
    dmm.setType(Type_DMM);
    dmm.setTitre(Titre_DMM);
    dmm.setVolume(Volume_DMM);
    ... //etc.

    Patient patient = new Patient();
    patient.setNumero(Numero);
    patient.setNom(Nom);
    patient.setPrenom(Prenom);
    ... //etc.

    Medecin medecin = new Medecin();
    Medecin.setNumero(Numero_Medecin);
    ... //etc.
    patient.setMedecin(medecin);

    Admission admission = new Admission();
    admission.setHeure(Heure_Admission);
    admission.setDate(Date_Admission);
    ... //etc.
    patient.setAdmission(admission);

    dmm.setPatient(patient);

    Vector examens = new Vector();
    dmm.setExamens(examens);

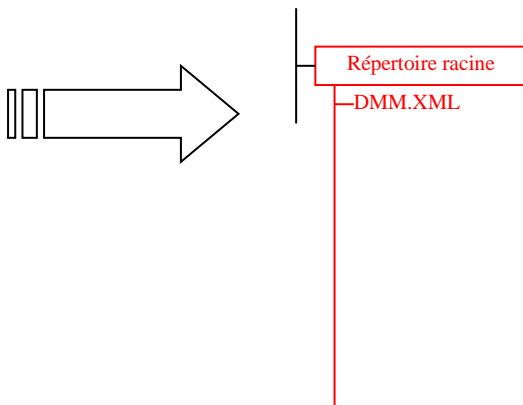
    //Composition du fichier DMM.xml
    //dans la racine spécifiée par Path
    Document doc = new Document(Path,dmm);
    ... //Ajout d'une page (2ème Etape).
  }
}

class Document
{ public String DocumentPathName = "";
  public DMM Dmm = new DMM();
  public Vector Emmch = new Vector();
  public Vector VectorScenarios = new Vector();

  public Document(String documentPathName, DMM dmm)
  { DocumentPathName = documentPathName;
    Dmm = dmm;
    try
    { //Créer le répertoire racine
      File dirRoot = new File(documentPathName);
      dirRoot.mkdirs();

      //Creation du fichier DMM.xml
      File file = new File(documentPathName + "DMM.xml");
      PrintWriter Output =
        new PrintWriter( new FileOutputStream(file) );

      //Ecriture de l'objet DMM dans le fichier
      DMMwriter writer = new DMMwriter();
      writer.writeDMM(Output,dmm);
      Output.flush();
      Output.close();
    }
    catch(Exception E)
    { System.out.println("Error in write DMM.XML");
    }
  }
}
```



2^{ème} Etape : Ajout d'une page au document (avec ou sans éléments multimédia)

- Ajout d'une page **sans** élément multimédia

```

class Composition
{ public void forDefault( paramètres )
  { ... //(1ère Etape).
    //Composition d'un objet EMMCH en mémoire
    EMMCH page = new EMMCH();
    page.setType(Type_EMMCH);
    page.setTitre(Titre_EMMCH);
    ... //etc.

    Contexte contexte = new Contexte();
    contexte.setHeure(Heure_Prelevement);
    contexte.setDate(Date_Prelevement);
    ... //etc.
    page.setContexte(contexte);

    Enregistrement enregistrement
      = new Enregistrement();
    enregistrement.setNumero(Numero_Enregistrement);
    enregistrement.setHeure(Heure_Enregistrement);
    ... //etc.
    page.setEnregistrement(enregistrement);

    Vector prescripteurs = new Vector();
    page.setPrescripteurs(prescripteurs);

    //Composition du vecteur de références aux
    //scénarios utilisés
    Vector Scenarios = new Vector();
    Scenarios
      .addElement("c:\\... \\Presentation1\\Scenario.xml");
    Scenarios
      .addElement("c:\\... \\Presentation2\\Scenario.xml");
    Scenarios
      .addElement("c:\\... \\Presentation3\\Scenario.xml");

    //Composition de la page
    doc.addPage(page, Scenarios);
    document = doc;
  }
}

```

```

public class Document
{ public void addPage(EMMCH emmch, Vector scenarios)
  { //Créer le répertoire de la page
    File dir_Page = new File( DocumentPathName + "Examen"
      + numberOfpage);
    dir_Page.mkdirs();

    //Créer le répertoire EM dans le répertoire racine
    File dir_EM = new File( DocumentPathName + "EM" );
    dir_EM.mkdirs();

    Vector medias = new Vector();
    emmch.setMedias(medias);

    ... //Ajout des Scénarios (page suivante)

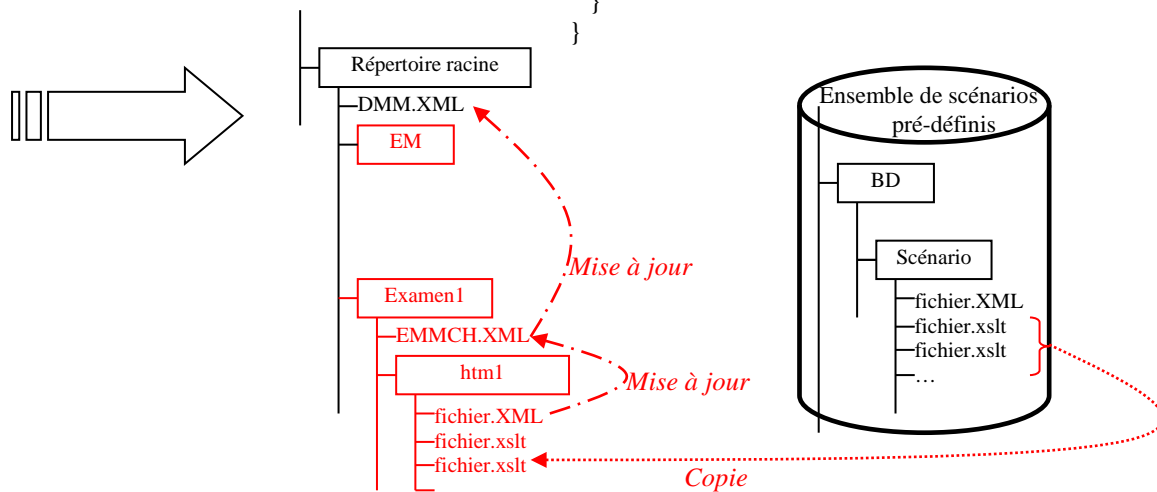
    //Ecriture de l'objet EMMCH dans le fichier
    //EMMCH.xml
    File file_page = new File(dir_Page + "\\EMMCH.xml");
    PrintWriter Output
      = new PrintWriter( new FileOutputStream(file_page) );
    emmch.setEmplacement("Examen" + numberOfpage
      + "\\EMMCH.xml");
    EMMCHwriter writer = new EMMCHwriter();
    writer.writeEMMCH(Output,emmch);

    //Ajouter l'examen (page) à l'objet DMM
    Examen examen = new Examen();
    examen.setType(emmch.getType());
    ... //etc.
    Dmm.getExamens().addElement(examen);

    //Mise à jour du fichier DMM.xml
    File file = new File(DocumentPathName + "DMM.xml");
    PrintWriter Output
      = new PrintWriter( new FileOutputStream(file) );
    DMMwriter writer = new DMMwriter();
    writer.writeDMM(Output,Dmm);

    Emmch.addElement(emmch);
    VectorScenarios.addElement(scenari);
  }
}

```



//Ajout des Scénarios

```
int index = 0;
Vector scenari = new Vector();
while ( index < scenarios.size() )
{ String scenarioPath = (String) scenarios.elementAt(index);
  File file_scenario = new File( scenarioPath );

  //Lire le fichier Scenario.xml contenu dans un répertoire base de données afin de récupérer un objet Scenario
  //correspondant à l'information contenue dans le fichier
  InputStreamReader Input = new InputStreamReader(new BufferedInputStream( new FileInputStream(file_scenario) ) );
  Scenarioreader DocReader = new Scenarioreader();
  Scenario scenario = DocReader.readScenario(Input);

  //Créer le répertoire du scénario
  String Acces = scenario.getAcces();
  File dir_scenario = new File( dir_Page + "\\" + Acces );
  dir_scenario.mkdirs();

  //Mise à jour du scénario
  scenario.setEmplacement( "Examen" + String.valueOf(numberOfpage) + "\\" + Acces + "\\Scenario.xml" );
  scenario.setMedias(medias);
  Vector transformations = scenario.getTransformations();
  int indexTransformation = 0;
  while ( indexTransformation < transformations.size() )
  { Transformation transformation = (Transformation) transformations.elementAt(indexTransformation);
    Vector Inputs = transformation.getInputs();
    int indexInputs = 0;
    while (indexInputs < Inputs.size())
    { Vector input = (Vector)Inputs.elementAt(indexInputs);
      String InputSource = (String) input.elementAt(1);
      if ( InputSource.compareTo("DMM.xml")==0 ) { //do nothing } else
      { if ( InputSource.compareTo("EMMCH.xml")==0 )
        { input.setElementAt( "Examen" + String.valueOf(numberOfpage) + File.separator+"EMMCH.xml", 1);
        } else
        { if ( InputSource.startsWith("EM"+File.separator) ) { //do nothing } else
          { input.setElementAt("Examen"+String.valueOf(numberOfpage)+File.separator+
            Acces+File.separator+"Manifestation"+File.separator+InputSource, 1);
          };
        };
      };
    };
    indexInputs = indexInputs + 1;
  };
  File InputFile = new File( file_scenario.getParent() + "\\" + transformation.getEmplacement_Feuille() );
  File OutputFile = new File(dir_Page + "\\" + Acces + "\\" + transformation.getEmplacement_Feuille() );
  transformation.setEmplacement_Feuille( "Examen" + numberOfpage + "\\" + Acces + "\\" +
    transformation.getEmplacement_Feuille() );

  //Copie des feuilles de transformation dans le répertoire
  FileReader in = new FileReader(InputFile);
  FileWriter out = new FileWriter(OutputFile);
  int character; while ((character = in.read()) != -1) out.write(character);
  indexTransformation = indexTransformation + 1;
}
//Creation du fichier Scenario.xml
//Ecriture de l'objet scenario dans le fichier Scenario.xml
File file = new File(dir_Page + "\\" + Acces + "\\Scenario.xml");
PrintWriter Output = new PrintWriter( new BufferedOutputStream( new FileOutputStream(file) ) );
Scenariowriter writer = new Scenariowriter();
writer.writeScenario(Output,scenario);
scenari.addElement(scenario);
index = index+1;
};
emmch.setPresentations(scenari);
```

- Ajout d'une page avec éléments multimédias

```

class Composition
{ public void forDefault( paramètres )
{ ... //(1ère Etape).
//Composition d'un objet EMMCH en mémoire
EMMCH page = new EMMCH();
page.setType(Type_EMMCH);
page.setTitre(Titre_EMMCH);
... //etc.

Contexte contexte = new Contexte();
contexte.setHeure(Heure_Prelevement);
contexte.setDate(Date_Prelevement);
... //etc.
page.setContexte(contexte);

Enregistrement enregistrement
= new Enregistrement();
enregistrement.setNumero(Numero_Enregistrement);
enregistrement.setHeure(Heure_Enregistrement);
... //etc.
page.setEnregistrement(enregistrement);

Vector prescripteurs = new Vector();
page.setPrescripteurs(prescripteurs);

//Composition du vecteur de références aux
//scénarios utilisés
Vector Scenarios = new Vector();
Scenarios
.addElement("c:\\... \\Presentation1\\Scenario.xml");
Scenarios
.addElement("c:\\... \\Presentation2\\Scenario.xml");
Scenarios
.addElement("c:\\... \\Presentation3\\Scenario.xml");

//Composition de la page
doc.addPage(page, Dir_Medias, Scenarios,
new MediaTypeDefinerEMMCH() );
document = doc;
}
}

```

```

public class Document
{ public void addPage(EMMCH emmch, Vector scenarios)
{ //Créer le répertoire de la page
File dir_Page = new File( DocumentPathName + "Examen"
+ numberOfpage);
dir_Page.mkdirs();

//Créer le répertoire EM dans le répertoire racine
File dir_EM = new File( DocumentPathName + "EM" );
dir_EM.mkdirs();

//Composition de la liste des medias associés à l'examen
//Lister les fichiers contenus dans la directorie d'acquisition
File dir_medias = new File(dir_medias_path);
String[] FilesPath = dir_medias.list();
int index = 0;
Vector medias = new Vector();
while ( index < FilesPath.length )
{ File file_media = new File(dir_medias_path +
FilesPath[index]);

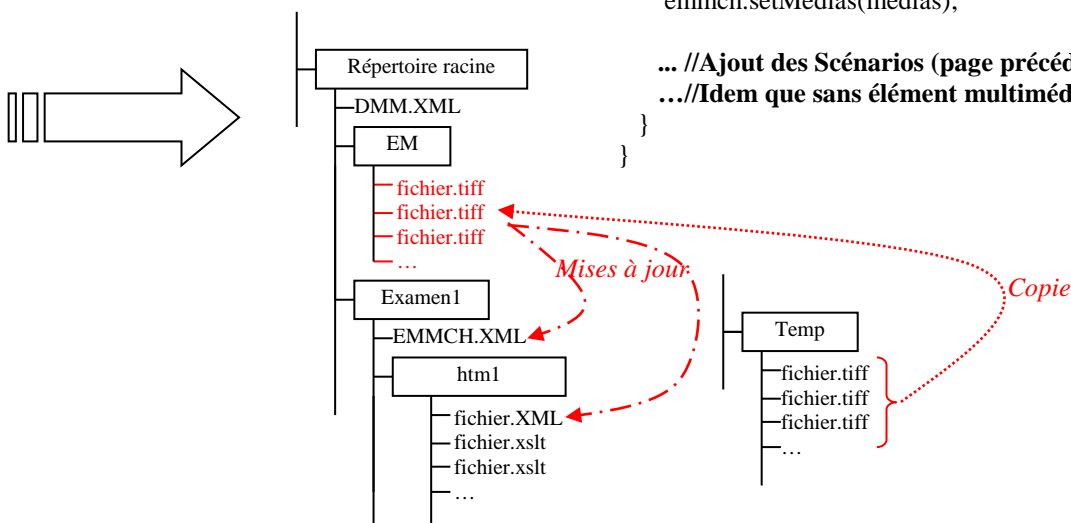
if ( file_media.isFile() )
{ long size = file_media.length();
long epoch = file_media.lastModified();

//Créer un objet Media pour chaque média
Media media = new Media();
media.setType( mtd.getTypeOfMedia(FilesPath[index]) );
media.setVolume( String.valueOf(size) );
media.setNom( FilesPath[index] );
media.setVersion("");
media.setEmplacement( "EM\\" + FilesPath[index]);
... //etc.
medias.addElement(media);

//Déplacer le fichier media vers le répertoire EM
file_media.renameTo(new File(dir_EM + "\\" +
FilesPath[index]));
};
index = index + 1;
};

//Ajouter les objets Media à la page
emmch.setMedias(medias);

```



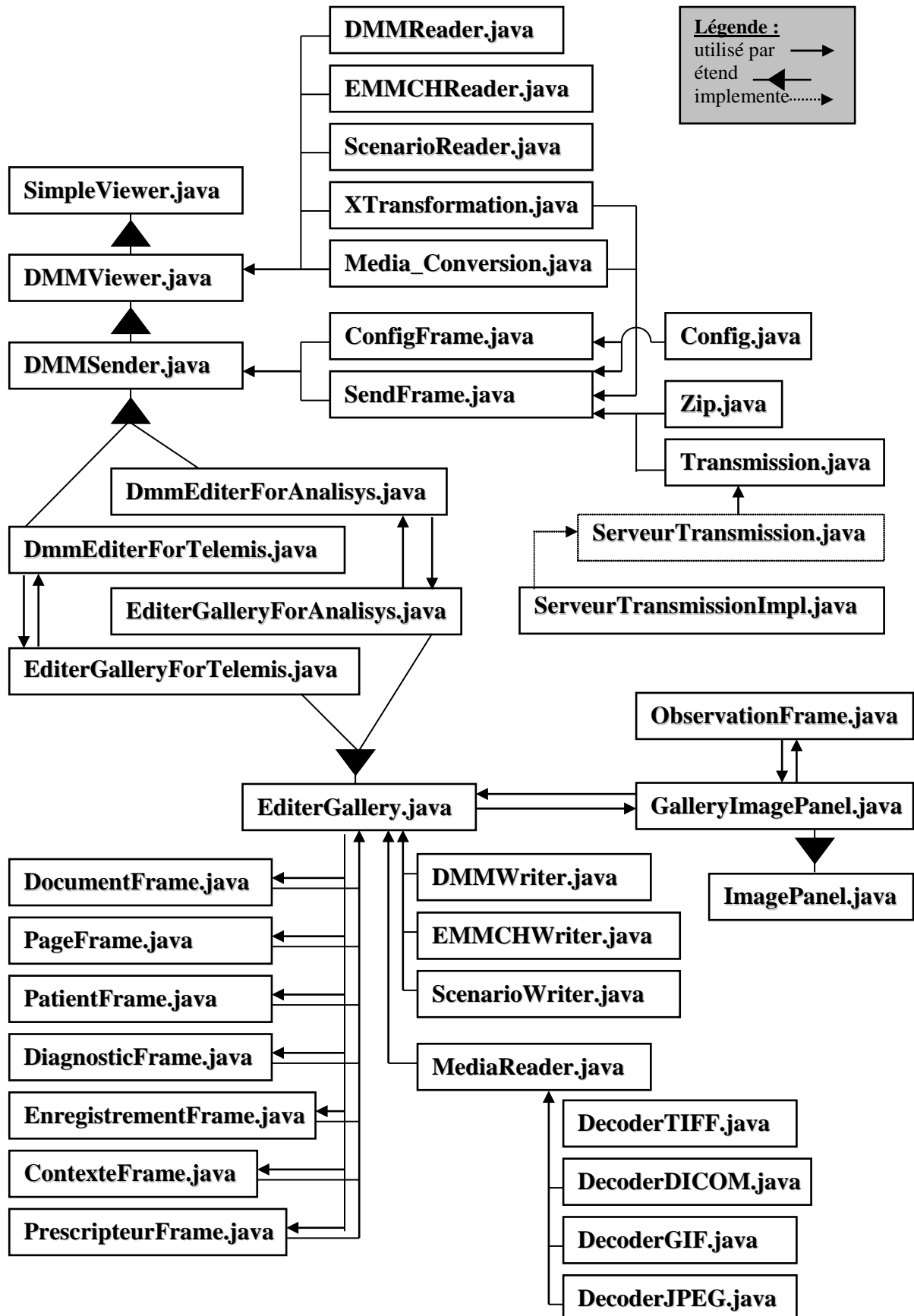
```

... //Ajout des Scénarios (page précédente)
... //Idem que sans élément multimédia
}
}

```

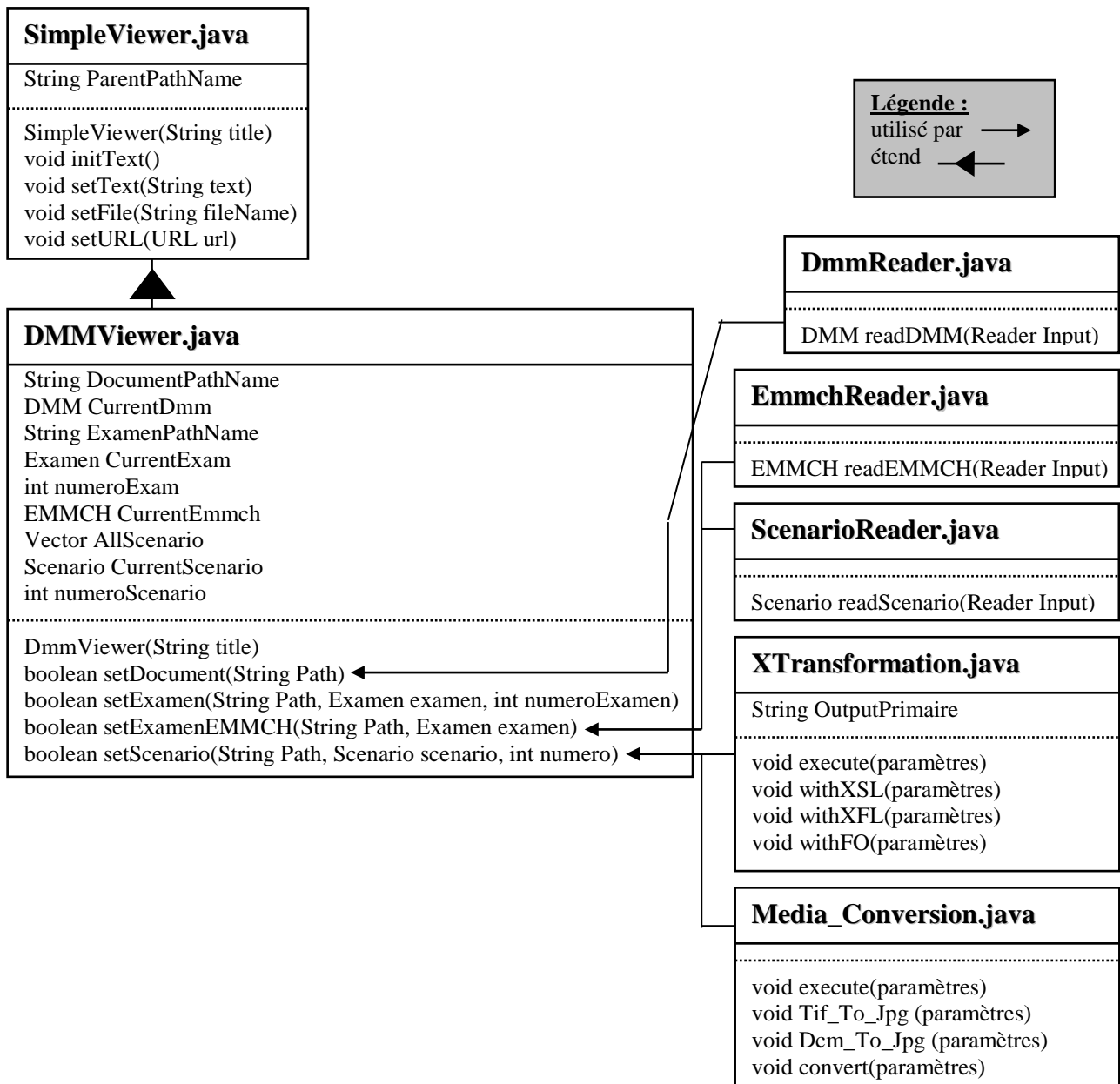
Annexe 17 : La visualisation, l'édition et la transmission d'un document multimédia médical en JAVA

Principales classes utilisées



Visualisation d'un document multimédia médical

Principales classes utilisées



1^{ère} Etape : Lire la première page du document

```
class DMMViewer
{ public boolean setDocument(String Path)
{ ...
//S'assurer que le fichier DMM.xml existe dans le répertoire racine du document
if ( ( new File( Path + "DMM.xml" ).exists() ) ) { DocumentPathName = Path ; }
else
{ //Si le fichier DMM.xml n'existe pas dans ce répertoire, alors le document est considéré comme mal formaté
//Initialiser un document vide.
initText() ;
//Afficher un message d'erreur.
JOptionPane.showMessageDialog(this, "This is not a well-formed EMIM document", "Can't open this
document", JOptionPane.WARNING_MESSAGE) ;

return false ;
} ;

//Si le fichier DMM.xml existe dans ce répertoire.
//Lire le fichier DMM.xml afin de récupérer un objet DMM correspondant à l'information contenue dans le fichier
File file = new File( Path + "DMM.xml" ) ;
InputStreamReader Input = new InputStreamReader( new BufferedInputStream( new FileInputStream(file) ) ) ;
DMMreader DocReader = new DMMreader() ;
CurrentDmm = DocReader.readDMM(Input) ;
Input.close() ;

//Récupérer la liste (vecteur) des examens contenus dans le document.
Vector examens = CurrentDmm.getExamens() ;
...

//Lire le premier examen de cette liste.
Examen examen = (Examen) examens.elementAt(0) ;
return setExamen( Path, examen, 0 ) ;
}

public boolean setExamen( String Path, Examen examen, int numeroExamen )
{ ...

String Type = examen.getType();
CurrentExam = examen;
numeroExam = numeroExamen;
...
return setExamenEMMCH( Path, examen ) ;
}

public boolean setExamenEMMCH(String Path, Examen examen)
{ ...

//S'assurer que le fichier Page.xml existe
if ( ( new File( Path + examen.getEmplacement() ).exists() ) )
{ ExamenPathName = Path + examen.getEmplacement(); }
else
{ //Si le fichier Page.xml n'existe pas
//Initialiser un document vide.
initText();
//Afficher un message d'erreur
JOptionPane.showMessageDialog(this, "This is not a well-formed EMIM document", "Can't open this
document", JOptionPane.WARNING_MESSAGE);

return false;
};
};
```

```

//Si le fichier Page.xml existe
//Lire le fichier Page.xml afin de récupérer un objet EMMCH correspondant à l'information contenue dans le fichier
File file = new File( Path + examen.getEmplacement() );
InputStreamReader Input = new InputStreamReader( new BufferedInputStream( new FileInputStream(file) ) );
EMMCHreader DocReader = new EMMCHreader();
CurrentEmmch = DocReader.readEMMCH(Input);
Input.close();

//Récupérer la liste (vecteur) des présentations contenues dans la page
Vector presentations = CurrentEmmch.getPresentations();
...
//Lire les fichiers Scenario.xml afin de récupérer des objets Scenario correspondant à l'information contenue dans
//ces fichiers
int index = 0;
while( index < presentations.size() )
{ File file = new File( Path + presentation.getEmplacement() );
  InputStreamReader Input = new InputStreamReader( new BufferedInputStream( new FileInputStream(file) ) );
  Scenarioreader DocReader = new Scenarioreader();
  Scenario scenario = DocReader.readScenario(Input);
  Input.close();
  AllScenario.addElement(scenario);
  index = index + 1;
}
... //(2ème Etape)
}
}

```

2^{ème} Etape : Afficher le première scénario de la page

```

class DMMViewer
{ ...

public boolean setExamenEMMCH(String Path, Examen examen)
{
  ...//(1ère Etape)

  //Afficher le premier scénario de cette liste
  Scenario scenario = (Scenario) AllScenario.elementAt(0);
  return setScenario( Path, scenario, 0 );
}

public boolean setScenario( String Path, Scenario scenario, int numero )
{ String FileHtmlPath = "";
  CurrentScenario = scenario;
  numeroScenario = numero;

  //Exécuter le scénario
  //Exécuter les transformations
  XTransformation transformation = new XTransformation();
  transformation.execute( Path, scenario );
  //Exécuter les conversions
  Media_Conversion conversion = new Media_Conversion();
  conversion.execute( Path, scenario );
  FileHtmlPath = Path + transformation.OutputPrimaire;

  //Afficher la manifestation correspondant à l'exécution du scénario
  setFile( FileHtmlPath );
  return true;
}
}

```

- Exécuter les transformations

```

class XTransformation
{
...
public void execute ( String Path, Scenario scenario )
{ //Créer un répertoire Manifestation dans le répertoire du scénario
  Dir_Manifestation = Dir_Scenario + "Manifestation"+File.separator;
  File file = new File( Path + Dir_Manifestation );
  file.mkdirs();
  //Pour chaque transformation du scénario choisir le processeur qui convient (XSLT, XFL, FO)
  Vector transformations = scenario.getTransformations();
  int index = 0; while( index < transformations.size() )
  { Transformation element = (Transformation) transformations.elementAt(index); index = index + 1;
    if ( element.getType_Feuille().compareTo("xsl")==0 ) { withXSL(Path,element); };
    if ( element.getType_Feuille().compareTo("xfl")==0 ) { withXFL(Path,element); };
    if ( element.getType_Feuille().compareTo("fo")==0 ) { withFO(Path,element); };
  }
}

public void withXSL( String Path, Transformation element)
{ FileOutputStream Output;
  if (element.getType_Output().equals("primaire"))
  { OutputPrimaire=Dir_Manifestation+element.getEmplacement_Output(); };
  //Créer le fichier output et les différents sous-répertoires
  File File_Output = new File( Path + Dir_Manifestation + element.getEmplacement_Output() );
  File Dir_Output = new File( File_Output.getParent() );
  Dir_Output.mkdirs();
  Output = new FileOutputStream( File_Output );
  //Initialiser le processeur avec la feuille de style
  TransformerFactory tFactory = TransformerFactory.newInstance();
  File ft = new File( Path + element.getEmplacement_Feuille() );
  Transformer transformer = tFactory.newTransformer(new StreamSource( ft ));
  ...
  //Exécuter la transformation
  transformer.transform(new StreamSource(new File( Path + primaryInput ) ), new StreamResult(Output));
  ...
}

public void withXFL( String Path, Transformation element)
{ FileWriter Output;
  if ( element.getType_Output().equals("primaire") )
  { OutputPrimaire = Dir_Manifestation + element.getEmplacement_Output(); };
  //Créer le fichier output et les différents sous-répertoires
  File File_Output = new File( Path + Dir_Manifestation + element.getEmplacement_Output() );
  File Dir_Output = new File( File_Output.getParent() );
  Dir_Output.mkdirs();
  Output = new FileWriter( File_Output );
  //Initialiser le processeur avec la feuille de style
  XmlConvert xml_convert = new XmlConvert(new FileReader(Path+element.getEmplacement_Feuille() ),true) ;
  ...
  //Exécuter la transformation
  xml_convert.flatToXml( new FileReader( Path + primaryInput ), Output );
}

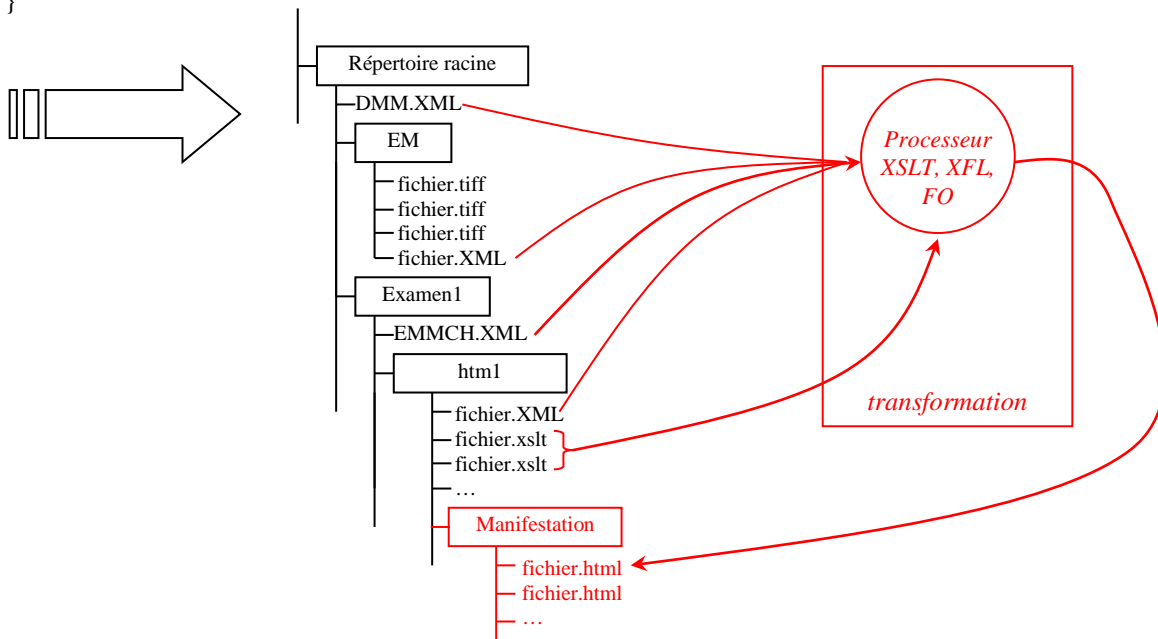
```



```

public void withFO( String Path, Transformation element)
{ File File_Output;
  if ( element.getType_Output().equals("primaire") )
  { OutputPrimaire = Dir_Manifestation + element.getEmplacement_Output(); };
  //Créer le fichier output et les différents sous-répertoires
  File_Output = new File( Path + Dir_Manifestation + element.getEmplacement_Output() );
  File Dir_Output = new File( File_Output.getParent() );
  Dir_Output.mkdirs();
  ...
  String[] arg = new String[2]; arg[0] = Path + primaryInput; arg[1] = File_Output.getPath();
  //Exécuter la transformation
  CommandLineOptions options = new CommandLineOptions(arg);
  new MyCommandLineStarter().run(options);
  ...
}
}

```



- Exécuter les conversions

```

public class Media_Conversion
{ ...
  public void execute ( String Path, Scenario scenario )
  { ...
    //Créer un répertoire Manifestation dans le répertoire du scénario s'il ne l'est pas.
    Dir_Manifestation = Dir_Scenario + "Manifestation"+File.separator;
    File file = new File( Path + Dir_Manifestation );
    file.mkdirs();

    //Créer un répertoire EM dans le répertoire de la manifestation
    Dir_EM = Dir_Manifestation + "EM"+File.separator;
    file = new File( Path + Dir_EM );
    file.mkdirs();

    Vector Conversions = scenario.getConversions();
    Vector Medias = scenario.getMedias();
    int index_media = 0;
    while (index_media < Medias.size()) //Pour chaque média contenu dans le scénario
    { Media media = (Media)Medias.elementAt(index_media);
      String media_type_in = media.getType();
      String media_type_out = media.getType();
      String compression = "no";
    }
  }
}

```

```

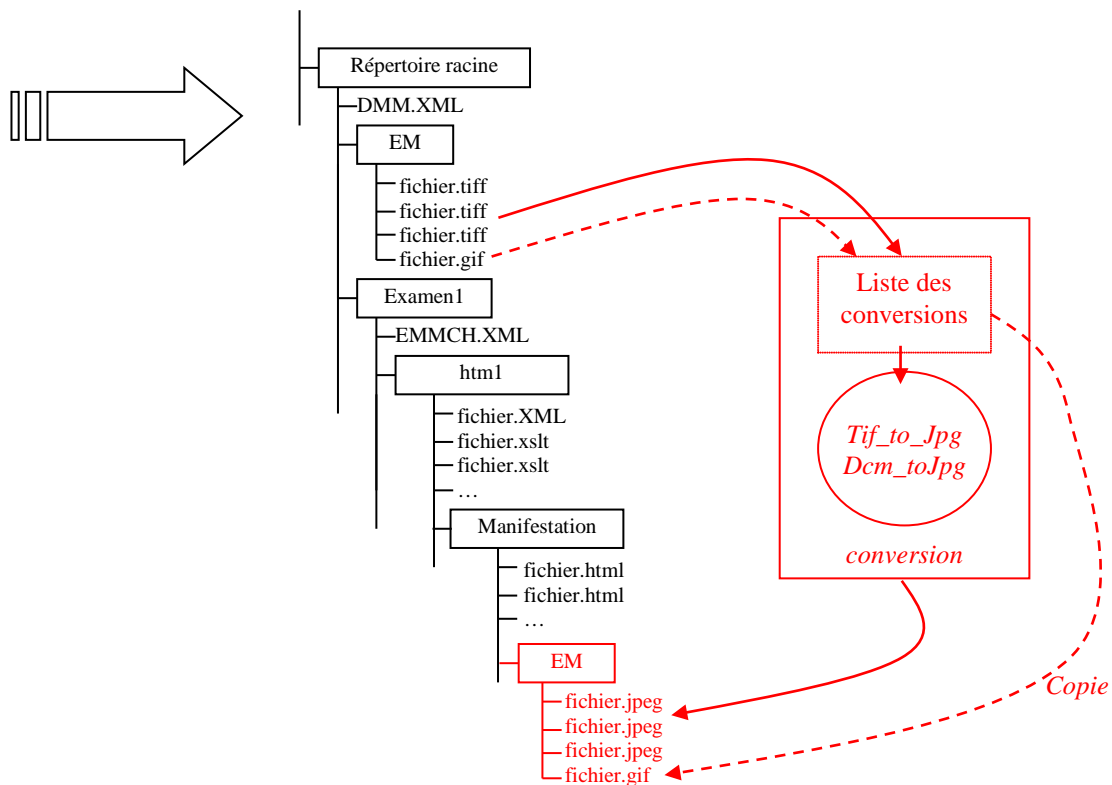
//Regarder s'il y a une conversion à effectuer pour le média
int index_conversion = 0;
while (index_conversion < Conversions.size())
{ Conversion conversion = (Conversion) Conversions.elementAt(index_conversion);
  if (media_type_in.equals(conversion.getType_In()))
  { media_type_out = conversion.getType_Out();
    compression = conversion.getCompression();
  }
  index_conversion = index_conversion + 1;
}
//S'il y a une conversion à effectuer, alors exécuter cette conversion et copier le média dans le répertoire EM
if ( ! media_type_in.equals(media_type_out) )
{ convert( Path, scenario, media, media_type_in, media_type_out, compression ); }
//Sinon copier le média dans le répertoire EM
else
{ File InputFile = new File( Path + media.getEmplacement() );
  File OutputFile = new File(Path + Dir_Manifestation + media.getEmplacement() );
  MyUtil.copyFile( InputFile.getPath() , OutputFile.getParent() , OutputFile.getName() );
}
index_media = index_media + 1;
}
}
}

```

```

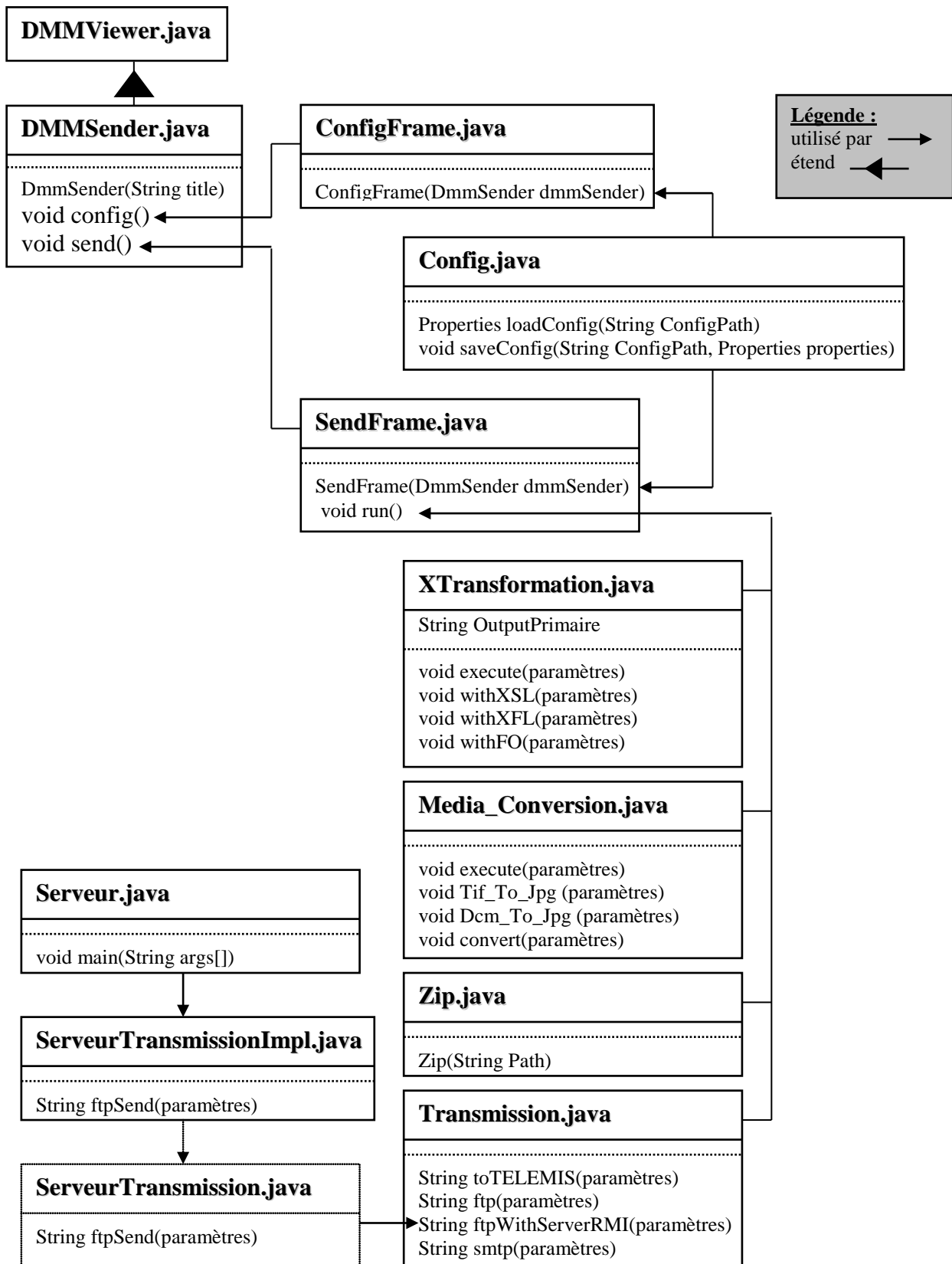
public void convert( String Path, Scenario scenario, Media media, String type_in, String type_out, String
compression )
{ if ( type_in.equals("Image/Tif") && type_out.equals("Image/Jpg") )
  { ... Tif_To_Jpg( input, output, compression ); return; };
  if ( type_in.equals("Image/Dcm") && type_out.equals("Image/Jpg") )
  { ... Dcm_To_Jpg( input, output, compression ); return; };
}
...
}

```



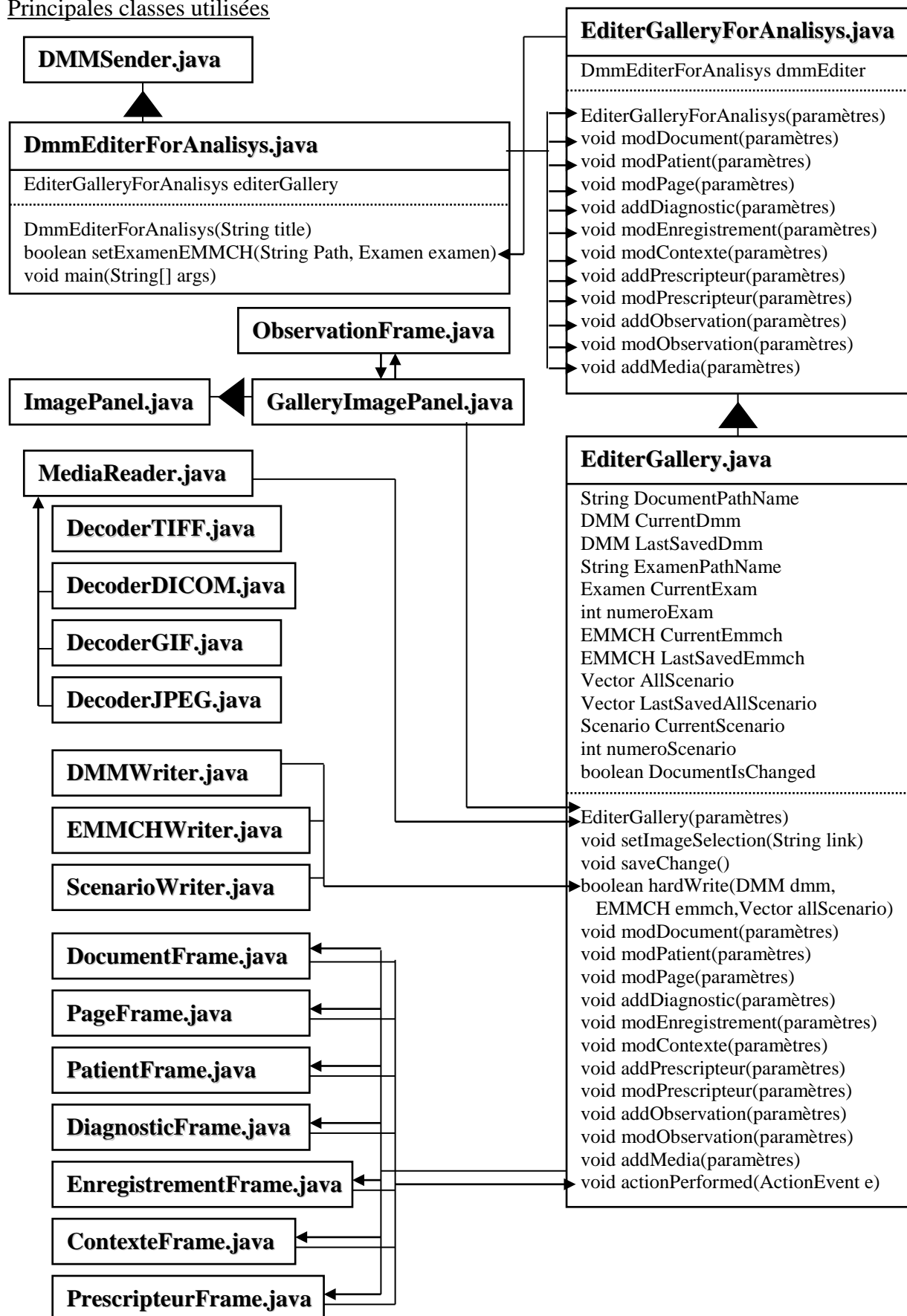
Transmission d'un document multimédia médical

Principales classes utilisées



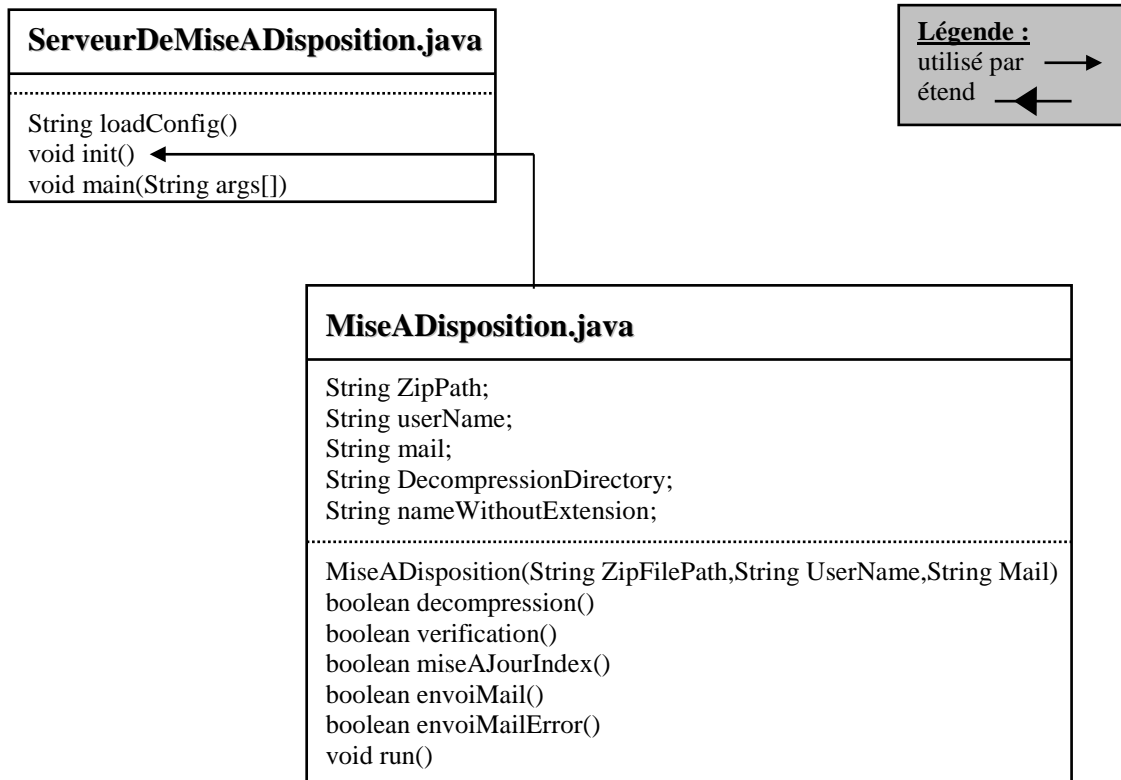
Edition d'un document multimédia médical

Principales classes utilisées



Annexe 18 : Le serveur de mise à disposition

Principales classes utilisées



1^{ère} Etape : Configuration du serveur

```
public class ServeurDeMiseADisposition
{ public static final String ConfigFilePath = "."+File.separator+"ServerConfig.ini";
  public static Vector Users = new Vector();

  public static void main(String args[])
  { ServeurDeMiseADisposition server = new ServeurDeMiseADisposition();
    server.init();
  }

  public void init()
  { //Configuration
    String state = loadConfig();
    if (state.equals("ok"))
    { System.out.println("Server ready !");
      ... // (2ème Etape)
    }
  }


  public static String loadConfig()
  { ...
    //Lecture du fichier de configuration
    File file = new File( ConfigFilePath );
    BufferedReader Input = new BufferedReader(new FileReader( file ));
    String line;
    while ((line = Input.readLine()) != null)
    { StringTokenizer stkz = new StringTokenizer(line,":");
      String UserName = new String(stkz.nextToken());
      String Password = new String(stkz.nextToken());
      String Group = new String(stkz.nextToken());
      String X1 = new String(stkz.nextToken());
      String Mail = new String(stkz.nextToken());
      String Directory = new String(stkz.nextToken());
      String X2 = new String(stkz.nextToken());
      User user = new User(UserName,Mail,Directory);
      Users.add(user);
    };
    Input.close();
    ...
    //Vérification de l'existence du répertoire public_html pour chaque compte d'utilisateur
    int i = 0;
    while (i < Users.size())
    { String directoryPath = ((User)Users.get(i)).directory;
      File directoryFile = new File( directoryPath );
      if (directoryFile.exists())
      { if (directoryFile.isDirectory())
        { File htmlDirectoryFile = new File(directoryFile,"public_html");
          if (htmlDirectoryFile.mkdir())
          { MyUtil.copyFile( "."+ File.separator + "index.html", htmlDirectoryFile.getPath(), "index.html" ); }
        } else
        { return "Config Directory (" +directoryPath+" ) isn't a valid directory";
        }
      } else
      { return "Config Directory (" +directoryPath+" ) doesn't exist";
      }
      i = i + 1;
    }
    return "ok";
  }
}
```

2^{ème} Etape : Inspection continue des répertoires et mise à disposition des documents trouvés

```
public class ServeurDeMiseADisposition implements FilenameFilter
{
    ...
    public void init()
    {
        ... (1ère Etape)
        while (!arret)
        {
            //Recherche des fichiers « zip » pour chaque compte utilisateur
            int indexUser = 0;
            while (indexUser < Users.size())
            {
                User user = (User) Users.get(indexUser);
                File directoryFile = new File( user.directory );
                File htmlDirectoryFile = new File(directoryFile,"public_html");
                String[] listOfZipFile = directoryFile.list(this);
                //Pour un compte, mettre à disposition tous les fichiers « zip » trouvés
                int i = 0;
                while( i < listOfZipFile.length )
                {
                    ...
                    File ZipFileIn = new File(directoryFile,listOfZipFile[i]); ...
                    File ZipFileOut = new File(htmlDirectoryFile,ZipFileIn.getName());
                    ...
                    if ( ZipFileIn.renameTo(ZipFileOut) )
                    {
                        System.out.println("Mise A Disposition : "+ZipFileOut.getPath()+"/"+user.userName+"/"+user.mail);
                        MiseADisposition process = new MiseADisposition(ZipFileOut.getPath(),user.userName,user.mail);
                    }
                    ...
                    i = i + 1;
                }
                indexUser = indexUser + 1;
            }
        }
    }
}

public class MiseADisposition implements Runnable
{
    ...
    Thread process;
    String ZipPath;
    String userName;
    String mail;
    String DecompressionDirectory;
    String nameWithoutExtension;
    ...

    public MiseADisposition(String ZipFilePath,String UserName,String Mail)
    {
        ZipPath = ZipFilePath;
        userName = UserName;
        mail = Mail;
        DecompressionDirectory = new File(ZipFilePath).getParent();
        String name = new File(ZipPath).getName();
        nameWithoutExtension = name.substring( 0 , name.lastIndexOf(".") );
        process = new Thread(this);
        process.start();
    }
}
```

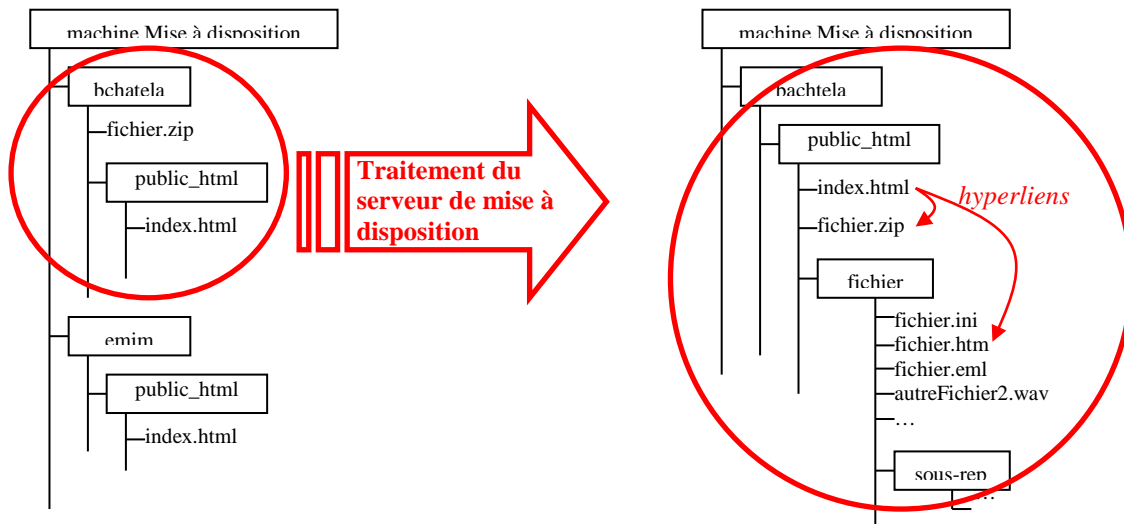


```

public class MiseADisposition implements Runnable
{
    ...
    public void run()
    {
        boolean ok = decompression();
        if (ok)
        {
            ok = verification();
            if (ok)
            {
                miseAJourIndex();
                envoiMail();
            }
        }
        else
        {
            envoiMailError();
            File fileToDelete = new File(ZipPath);
            fileToDelete.delete();
            MyUtil.removeDir(DecompressionDirectory + File.separator + nameWithoutExtension);
        }
    }
}
else
{
    envoiMailError();
    File fileToDelete = new File(ZipPath);
    fileToDelete.delete();
    MyUtil.removeDir(DecompressionDirectory + File.separator + nameWithoutExtension);
}
}
}

```

Gestion des fichiers sur le serveur de mise à disposition



fichier ServeurConfig.ini

nom : mot de passe : ... : ... : mail : répertoire : ...

```

emim:x:1002:10:hubert.meurisse@fundp.ac.be:/emim:/bin/sh
bernard_chatelain:x:1003:10:bchatela@onco.ucl.mont.ac.be:/bchatela:/bin/sh

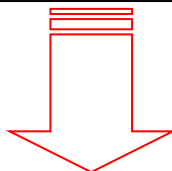
```


fichier index.html
(liste des documents mis à disposition)

```
<HTML><BODY>
<H1><U>LISTE DES DOCUMENTS DISPONIBLES</U></H1>
<TABLE border="5">
<TR>
  <TD><CENTER><FONT SIZE="3"><U>DOCUMENT</U></FONT></CENTER></TD>
  <TD><CENTER><FONT SIZE="3"><U>NOM DU PATIENT</U></FONT></CENTER></TD>
  <TD><CENTER><FONT SIZE="3"><U>PRENOM DU PATIENT</U></FONT></CENTER></TD>
  <TD><CENTER><FONT SIZE="3"><U>DATE D'EXAMEN</U></FONT></CENTER></TD>
  <TD><CENTER><FONT SIZE="3"><U>NOM DE L'EXPEDITEUR</U></FONT></CENTER></TD>
  <TD><CENTER><FONT SIZE="3"><U>ADRESSE DE
                                L'EXPEDITEUR</U></FONT></CENTER></TD>
  <TD><CENTER><FONT SIZE="3"><U>DOCUMENT SOUS FORMAT
                                ZIP</U></FONT></CENTER></TD>
</TR>

<!-- Add Line Here -->

</TABLE>
</BODY></HTML>
```



fichier index.html
(liste des documents mis à disposition)

```
<HTML><BODY>
<H1><U>LISTE DES DOCUMENTS DISPONIBLES</U></H1>
<TABLE border="5">
<TR>
  <TD><CENTER><FONT SIZE="3"><U>DOCUMENT</U></FONT></CENTER></TD>
  <TD><CENTER><FONT SIZE="3"><U>NOM DU PATIENT</U></FONT></CENTER></TD>
  <TD><CENTER><FONT SIZE="3"><U>PRENOM DU PATIENT</U></FONT></CENTER></TD>
  <TD><CENTER><FONT SIZE="3"><U>DATE D'EXAMEN</U></FONT></CENTER></TD>
  <TD><CENTER><FONT SIZE="3"><U>NOM DE L'EXPEDITEUR</U></FONT></CENTER></TD>
  <TD><CENTER><FONT SIZE="3"><U>ADRESSE DE
                                L'EXPEDITEUR</U></FONT></CENTER></TD>
  <TD><CENTER><FONT SIZE="3"><U>DOCUMENT SOUS FORMAT
                                ZIP</U></FONT></CENTER></TD>
</TR>

<TR>
  <TD><A HREF="/.fichier/fichier.htm">fichier</A></TD>
  <TD>Georges</TD>
  <TD>Benoit</TD>
  <TD>09/03/02</TD>
  <TD>bernard_chatelain</TD>
  <TD><A HREF="mailto:bchatela@onco.ucl.mont.ac.be">bchatela@onco.ucl.mont.ac.be</A></TD>
  <TD><A HREF="/.fichier.zip">fichier.zip</A></TD>
</TR>

<!-- Add Line Here -->

</TABLE>
</BODY></HTML>
```

Facultés Universitaires Notre-Dame de la Paix, Namur
Institut d'Informatique
Année académique 2001-2002

La télémicroscopie en cytologie hématologique

par Benoit Georges

Mémoire présenté en vue de l'obtention
du grade de Maître en Informatique

Résumé :

Ce mémoire décrit une structure de fonctionnement des systèmes de télémicroscopie, qui comprend les étapes de sélection, d'acquisition et de stockage d'images, mais également, l'intégration de ces images et des données cliniques sous forme de document multimédia, la transmission, la visualisation et l'édition de ces documents. Ce document expose également une tentative d'application de cette structure dans un environnement précis de développement, qui est celui de la cytologie hématologique à la clinique universitaire de Mont-Godinne. Conjointement, il présente un modèle de document multimédia permettant l'intégration des images et des données cliniques. Ce modèle de document tente de s'inscrire dans un cadre plus large que celui de la télémicroscopie et de satisfaire aux besoins de l'ensemble des pratiques médicales. Il est remarquable, par la manière dont il sépare le contenu de sa présentation, en employant le concept de scénarios permettant de transformer ou présenter le contenu sous forme de différentes manifestations.

Mots clés :

Cytologie hématologique, Télémicroscopie, Image numérique, Document multimédia.

Abstract :

This memoir describes a working structure of telemicroscopy systems which include the images selection, acquisition and registering, but also, the assembling of this images and other clinical data in shape of multimedia documents, the visualisation and the authoring of this documents. Then, it is tried to apply this structure in a definite environment for developing which is the one of hematological cytology at the university hospital of Mont-Godinne. In the same time, it presents a model of multimedia documents to allow the images and data assembling. This model of documents tries to join a larger framework than the one of telemicroscopy and to comply with the needs of all the medical disciplines. The model is remarkable by the way with it separates the contents and the presentation, using the idea of scenario which allows to transform and present the contents in different kinds of manifestations.

Key words :

Hematological cytology, Telemicroscopy, Numerical Image, Multimedia document.

Remerciements

Je remercie Monsieur Jean Fichet, Directeur à l'Institut d'Informatique des Facultés Universitaires Notre-Dame de la Paix, qui a accepté d'être le promoteur de ce mémoire. Il a malheureusement été empêché par la maladie de suivre la fin de ce travail. Merci à Monsieur Jean-Paul Leclercq, Professeur à l'Institut d'Informatique des Facultés Universitaires Notre-Dame de la Paix, d'avoir pris le relais.

Je remercie tout particulièrement Monsieur Hubert Meurisse, Coordinateur des recherches dans le domaine de la visualisation et l'analyse d'image à l'UCL Mont-Godinne et aux des Facultés Universitaires Notre-Dame de la Paix, co-promoteur de ce travail. Je lui suis reconnaissant pour l'intérêt qu'il a suscité chez moi à propos de la télémédecine et plus particulièrement de la télémicroscopie. Je le remercie aussi pour l'aide qu'il m'a fournie en proposant une bibliographie de départ, en traçant les grandes lignes et en apportant une discussion critique au fur et à mesure de la progression du projet réalisé durant mon stage.

Je remercie également, pour leur disponibilité et leurs nombreux conseils, Messieurs Bernard Chatelain, Chef de laboratoire associé et cytologiste à l'UCL Mont-Godinne, et Yvan Cornet, cytologiste à l'UCL Mont-Godinne, dont l'expertise dans le domaine de la cytologie hématologique m'a été indispensable. Ils m'ont initié à l'utilisation des outils nécessaires à la numérisation, microscope, capteur, lames.

La confiance que Monsieur Chatelain m'a accordée en me demandant de présenter ma recherche lors d'un séminaire concernant la télémicroscopie à la clinique universitaire Saint-Luc à Bruxelles a été un précieux encouragement.

Je remercie aussi Bruno Piscaglia, Customer Services Manager, et Patrice Roulive, Research & Development Manager, tous deux de la S.A. « Télémis », pour leur collaboration à ce projet et pour leur disponibilité à répondre aux nombreuses questions que je leur ai soumises.

Je remercie Koen Waelput, Responsable gestion de qualité à la S.A. « Omnilabo », pour l'aide apportée dans la compréhension du fonctionnement du programme « Analysis » et pour les informations qu'il m'a fournies à propos de l'implémentation de nouveaux modules dans ce même programme.

Je remercie Jacques Rifon pour la relecture attentive et critique de ce mémoire.

Enfin je remercie toute ma famille, pour son soutien durant ces cinq années d'études.

Table des matières

INTRODUCTION.....	1
PREMIERE PARTIE : ETAT DE L'ART	5
1. LA CYTOLOGIE HEMATOLOGIQUE.....	6
1.1. La définition.....	6
1.2. La description d'une analyse.....	6
2. LA TELEMICROSCOPIE.....	7
2.1. La définition.....	7
2.2. Les avantages et inconvénients.....	7
2.3. Les types de systèmes.....	9
2.4. Le support élémentaire : l'image numérisée.....	10
2.4.1. L'affichage d'une image numérisée.....	10
2.4.2. Le stockage d'une image numérisée.....	11
2.5. La structure fonctionnelle d'un système.....	13
2.5.1. La sélection des images.....	13
2.5.2. L'acquisition des images.....	13
2.5.3. Le stockage des images.....	14
2.5.4. L'intégration des données cliniques sous forme de document.....	16
2.5.5. La transmission des documents.....	20
2.5.6. La visualisation, l'édition et l'interprétation des documents.....	23
DEUXIEME PARTIE : MATERIEL ET METHODE.....	25
3. LA TELEMICROSCOPIE A L'UCL MONT-GODINNE.....	26
3.1. L'infrastructure disponible.....	26
3.1.1. La station d'acquisition.....	26
3.1.2. Le logiciel « Analysis ».....	26
3.1.3. Le logiciel « Télémis ».....	27
3.1.4. L'ADICAP.....	29
3.2. La cytologie hématologique à l'UCL Mont-Godinne.....	32
3.2.1. La situation initiale.....	32
3.2.2. Les objectifs.....	33
3.3. Le type du système à développer.....	35
3.4. La structuration fonctionnelle du système à développer.....	36
3.4.1. La sélection des images.....	36
3.4.2. L'acquisition des images.....	36
3.4.3. Le stockage des images.....	37
3.4.4. L'intégration des données cliniques sous forme de document.....	38
3.4.5. La transmission des documents.....	40
3.4.6. La visualisation, l'édition et l'interprétation des documents.....	40
4. LES TECHNOLOGIES UTILISEES.....	41
4.1. XML.....	41
4.2. XSLT.....	42
4.3. DICOM.....	45
4.3.1. Le Standard DICOM.....	45
4.3.2. DICOM et la microscopie.....	47
4.3.3. La compatibilité entre DICOM et TIFF.....	48
TROISIEME PARTIE : RESULTATS.....	49
5. LA CONCEPTION DU MODELE DE DOCUMENT.....	50
6. LA CONCEPTION DU SYSTEME.....	52
6.1. La découpe en composant.....	52
6.2. Le fonctionnement du système.....	54
6.2.1. La station d'acquisition.....	54
6.2.2. Le « Télémis reception ».....	63
QUATRIEME PARTIE : DISCUSSION.....	65

7. LES LACUNES DU SYSTEME	66
8. LE SYSTEME « IDEAL »	67
CONCLUSION	71
REFERENCES.....	73
ANNEXES	77
Annexe 1 : Les méthodes de compressions sans perte	78
Annexe 2 : La méthode de compression avec perte, JPEG	80
Annexe 3 : Les types de cable pour la transmission de données.....	82
Annexe 4 : Les normes d'implémentation pour les réseaux LAN	83
Annexe 5 : Les réseaux publics	85
Annexe 6 : L'interface de « Analysis ».....	87
Annexe 7 : Les interfaces de « Télémis »	88
Annexe 8 : L'élaboration d'un format compatible avec TIFF et DICOM.....	89
Annexe 9 : Les DTD du modèle de document multimédia médical.....	92
Annexe 10 : Les scénarios réservés à l'examen de cytologie hématologique	95
Annexe 11 : La composition d'un document via « Analysis ».....	109
Annexe 12 : La visualisation, la transmission et l'édition d'un document via « Analysis »	110
Annexe 13 : L'analyse automatique de lames via « Analysis »	115
Annexe 14 : La composition, la visualisation, l'édition et la transmission d'un document via « Télémis »	116
Annexe 15 : La structure du modèle de document multimédia médical en JAVA.....	117
Annexe 16 : La composition d'un document multimédia médical en JAVA.....	118
Annexe 17 : La visualisation, l'édition et la transmission d'un document multimédia médical en JAVA.....	123
Annexe 18 : Le serveur de mise à disposition.....	132

Glossaire et abréviations

ADCCP (Advanced Data Communication Control Procedure)

Protocole de la couche de liaison de données qui a pour but de permettre la communication entre deux machines adjacentes, dans des réseaux point à point.

ADICAP (Association pour le Développement de l'Informatique en Cytologie et Anatomie Pathologique)

Association regroupant des experts en cytologie et en anatomopathologie. Cette association est impliquée dans la mise au point d'un système d'échanges de documents entre spécialistes en cytologie et en anatomie pathologique, qui porte son nom. (VOIR p. 29)

ALOHA

Protocole de gestion d'accès au canal dans un réseau à diffusion.

Analysis

Logiciel conçu pour l'acquisition, l'analyse, le traitement et l'archivage d'images. (VOIR p. 26)

ASCII (American Standard Code for Information Interchange)

Norme permettant de coder, de manière uniforme, sous forme de nombre, les lettres de l'alphabet anglais et américain, de manière à ce qu'une lettre reste toujours la même lettre quelle que soit la machine.

Autofocus

Système de mise au point automatique équipant une caméra afin d'obtenir des images nettes.

Bande passante

Espace de fréquence pour lequel des signaux sont transmis sans affaiblissement sur un média de transmission particulier.

Cellule nucléée

Cellule qui possède un noyau.

Contraste [d'une image]

Différence du niveau de gris entre deux points qui composent l'image.

CSMA/CD (Carrier Sense Multiple Access/Collision Detection)

Protocole de gestion d'accès au canal dans un réseau à diffusion.

Cytologie hématologique

Etude de la forme et de la structure des cellules de sang. (VOIR p. 6)

Cytomètre

Appareil automatique de routine des laboratoires d'hématologie, qui analyse, cellule par cellule, la composition d'un échantillon de sang afin de connaître la quantité et la qualité des éléments qu'on y trouve.

Débit [en télécommunication]

Quantité d'informations que peut fournir par unité de temps, une voie de communication.

Définition [d'une image]

Nombre de pixels sur la surface totale d'une image, c'est-à-dire le nombre de lignes multiplié par le nombre de colonnes.

DICOM (Digital Imaging and Communications in Medicine)

Standard permettant de promouvoir la communication d'images digitales, de faciliter le développement et l'expansion de systèmes d'archivage et de communication d'images qui puissent s'intégrer avec les autres systèmes d'informations hospitaliers. (VOIR p. 45)

DLL (Dynamic Linked Library)

Librairie de fonctions qui sont partagées entre plusieurs applications. Ces fonctions peuvent être appelées par n'importe quel programme et ceci, indépendamment du langage de programmation.

DQDB (Distributed Queue Dual Bus)

Protocole de gestion d'accès au canal dans un réseau à diffusion.

Document multimédia

Entité qui est constituée d'un ensemble d'éléments d'information de base ou d'objets média (texte, graphique, son, vidéo, image) reliés par des relations de différentes natures (relations de composition, spatiales, temporelles et de navigation).

DTD (Document Type Definition)

Définition d'un langage de marquage particulier qui respecte la syntaxe ou le méta-langage XML. Un document XML peut être valide selon une DTD, il est alors considéré comme conforme XML basé sur la DTD. (VOIR p. 41)

EMIM (Emission Multimédia en Imagerie Médicale)

Projet de recherche qui a pour objectif de réaliser un environnement partagé d'édition/présentation multimédia en support à l'élaboration du diagnostic médical. Le système d'édition/présentation développé devrait permettre à un ensemble de prestataires intervenant dans l'élaboration du diagnostic de générer des éléments diagnostiques et de créer, modifier et présenter un document multimédia médical.

Feuille de transformation

Concept qui désigne un moyen permettant de rassembler des règles de transformation applicables à un document et interprétables afin de réaliser une adaptation de celui-ci, comme par exemple une feuille de style XSLT.

Firewall

Dispositif qui filtre les flux de données entrant et sortant d'un réseau privé afin d'assurer une certaine sécurité. Généralement, le réseau privé est relié aux autres réseaux, de telle manière que les paquets de données doivent obligatoirement passer sur une passerelle sur laquelle est placé le dispositif.

FTP (File Transfer Protocol)

Protocole de transfert de fichiers.

Gbps (Gigabyte par seconde)**HDLC** (High-level Data Link Control)**HTML** (Hypertext Markup Language)

Langage de balisage qui est le plus répandu pour les documents basés Web.

HTTP (HyperText Transfer Protocol)

Protocole standard permettant d'accéder aux ressources du Web.

IMAP (Interactive Mail Access Protocol)

Protocole qui permet de récupérer le courrier électronique d'une boîte distante.

Intensité lumineuse [d'une image]

Attribut qui qualifie la luminosité d'une couleur (couleur claire ou foncée).

Interopérabilité

Propriété des systèmes leurs fournissant la possibilité de fonctionner entre eux.

Interprétation photométrique

Propriété d'une image numérisée, à niveau de gris, qui indique quelle doit être l'interprétation des valeurs numériques : 0 représente le blanc et la valeur maximale le noir, ou inversement.

IP (Internet Protocole)

Protocole standard de l'Internet qui permet de transporter les données tout au long d'un chemin d'une source à un destinataire.

Kb (Kilobyte)**Kbps** (Kilobyte par seconde)**Manifestation** [d'un document multimédia]

Concept qui désigne une présentation particulière d'une partie d'un document multimédia, qui est obtenue grâce à l'application d'un scénario qui intègre la dimension spatiale, temporelle et hypermédia à cette présentation.

Mbps (Megabyte par seconde)**MIME** (Multi-purpose Internet Mail Extensions)

Spécification décrivant les formats de messages multimédias sur l'Internet.

Monochrome (Pixel)

Pixel pouvant représenter une et une seule couleur mais dont l'intensité peut varier.

Morphologique

Relatif à la forme, à la structure.

Netteté [d'une image]

Propriété d'une image obtenue lorsque le contraste est maximum.

Numérisation [d'une image]

Représentation d'une image par une matrice bidimensionnelle de valeurs numériques. (VOIR p. 10)

OSI (Open Systems Interconnection)**Pathologie**

Science des causes, des symptômes et de l'évolution des maladies.

Pixel (Picture Element)

Point élémentaire constituant l'image.

POP3 (Post Office Protocol 3)

Protocole qui permet de récupérer le courrier électronique d'une boîte distante.

Port série [connexion par]

Interface par laquelle l'ordinateur peut communiquer avec d'autres appareils et qui porte ce nom en raison du fait que les données sont envoyées bit après bit, en série. Ce port est très lent, mais très bon marché, et surtout absolument standard.

PPP (Point-to-Point Protocol)

Protocole de liaison de données point-à-point.

Profondeur de champ

Ecart de déplacement vertical de la platine d'un microscope pour lequel l'image reste nette.

Protocole

Règles et conventions utilisées pour la conversation entre deux machines. Cette conversation est organisée en couches. Chaque couche joue un rôle particulier dans la conversation et chaque couche possède ses propres protocoles.

Résolution [d'une image]

Nombre de pixels par unité de longueur.

Saturation [d'une image]

Attribut qui décrit l'altération d'une couleur pure avec le blanc ou un niveau de gris. Elle mesure la proportion de couleur pure par rapport au blanc. Elle représente le facteur de pureté de la couleur. Cette notion permet de distinguer le rose du rouge.

Scénario [d'un document multimédia]

Concept qui désigne le déroulement des transformations qu'il faut réaliser afin d'obtenir une manifestation d'un document multimédia.

SDLC (Synchronous Data Link Control)

Protocole de la couche de liaison de données qui a pour but de permettre la communication entre deux machines adjacentes, dans des réseaux point à point.

SGML (Standard Generalized Markup Language)

Langage de balisage utilisé comme base du langage HTML.

Shell

Interface à ligne de commandes ou surcouche logicielle d'un système d'exploitation qui interprète des commandes.

SLIP (Serial Line IP)

Protocole de liaison de données point-à-point.

SMTP (Simple Mail Transfer Protocol)

Protocole de transfert de courrier électronique.

Structuration fonctionnelle [d'un système informatique]

Décomposition, par raffinements successifs, d'une fonctionnalité du système à informatiser en une succession de traitements ou d'actions à entreprendre pour la réaliser.

TCP (Transmission Control Protocol)

Protocole de la couche de transport qui fournit plusieurs services dont le plus important est de permettre le transfert d'informations d'une machine émettrice à une machine réceptrice de manière fiable et économique, indépendante de la nature du ou des réseaux mis en place.

Teinte [d'une image]

Attribut qui décrit immédiatement une couleur pure, comme le rouge pur, le bleu pur, etc. La teinte définit la couleur elle-même.

Télémicroscopie

Pratique de la visualisation et du diagnostic, à distance, d'images en provenance d'un microscope. (VOIR p. 7)

Télémis

Logiciel de télémédecine permettant de consulter et d'échanger les images en provenance des plateaux techniques d'acquisition tels que la Magnetic Resonance, la Computed Tomography, la Positron Emission Tomography, etc. (VOIR p. 27)

UDP (User Data Protocol)

Protocole de la couche de transport qui fournit plusieurs services dont le plus important est de permettre le transfert d'informations d'une machine émettrice à une machine réceptrice de manière fiable et économique, indépendante de la nature du ou des réseaux mis en place.

URL (Uniform Resource Locator)

Format d'adressage universel pour désigner une ressource sur Internet.

XFL (XML Flat files Language)

Langage de balisage qui permet de décrire la structure et la syntaxe d'une classe de fichiers « plats ». Un document respectant ce langage et décrivant la structure d'une classe particulière de fichiers

« plats » peut être utilisé par « XML Convert », une application de conversion, afin de transformer un fichier plat de cette classe en un fichier XML.

XLink (XML Linking Language)

Langage de balisage qui définit des éléments pouvant être insérés dans un document XML et permettant de créer et de décrire des liens entre les ressources : parties du document XML ou autres documents.

XML (Extended Markup Language)

Syntaxe ou méta-langage qui fournit un cadre pour la définition de langages de balisage de document et de structuration de l'information. Un document XML est un document qui respecte un langage défini dans le cadre de XML. (VOIR p. 41)

XPath (XML Path Language)

Langage qui définit des expressions et des fonctions permettant d'atteindre les différentes parties d'un document XML et d'effectuer certaines manipulations sur celles-ci.

XSL (Extensible Stylesheet Language)

Langage qui permet d'exprimer des feuilles de styles, c'est-à-dire des fichiers décrivant comment présenter un document XML. Ce langage est composé de XSLT et XSL FO.

XSL FO (XSL Formatting Object)

Langage de balisage qui permet de décrire des dépendances spatiales entre des éléments d'informations et de décrire un format de présentation de document XML.

XSLT (XSL Transformation)

Langage de balisage qui permet de définir des règles de transformations destinées à transformer un document XML conforme basé sur une DTD en un autre. (VOIR p. 42)

WDM (Wave-length Division Multiple Acces)

Protocole de gestion d'accès au canal dans un réseau à diffusion.