Le système Rapace

RAPPORT DE TER MASTER 1 IFPRU

–

Vidéo Surveillance à Distance

Réalisé par :

- FERJANI Mohammed
- SERIAI Abderrahmane
- HENNANI Hakim
- SEDDIK Annes

Sous la direction de :

M. SERIAI Abdelhak-Djamel

Remerciements

C’est un agréable devoir d’exprimer nos profonds remerciements et reconnaissances à toutes les personnes qui nous ont apporté aide technique, soutien moral et qui nous ont permis d’acquérir l’esprit de travail.

Nous tenons à exprimer notre très profonde gratitude à notre encadrant, Monsieur SERIAI Abdelhak-Djamel, pour ses précieux conseils, sa disponibilité et son total dévouement pour donner naissance à ce travail.

Un remerciement spécial à Monsieur LECLERE Michel, responsable du Master1 Informatique, pour sa patience et sa gentillesse.

Nous adressons une pensé particulière à nos parents, pour leurs judicieux conseils et leur soutien sans faille tout au long de notre parcours universitaire.

Enfin, nos vifs remerciements sont adressés à tout le personnel de l’Université de Montpellier II, qui a fourni beaucoup d’effort et de patience pour nous préparer à réussir notre vie professionnelle.

Sommaire

1 INTRODUCTION

Notre groupe de TER Rapace se compose de 4 étudiants : FERJANI Mohamed, HENNANI Hakim, SEDDIK Annes et SERIAI Abderrahmane issu du parcours Génie Logiciel (GL) du M1 IFPRU de l'université de Montpellier 2. Ce TER consiste en la réalisation d'un système de vidéosurveillance à distance. La conduite de ce projet à été encadrée par Mr SERIAI Abdelhak-Djamel, maître de conférences à l'université Montpellier 2. Nous avons choisi ce TER car nous avons comme intérêt commun les nouvelles technologies, leurs applications concrètes dans la vie de tous les jours. Dans ce cadre, le développement d'un système de vidéosurveillance à distance, est une concrétisation de cet intérêt.

Le but du système qu'on a choisit de développer est de permettre à l'utilisateur de surveiller un ou plusieurs locaux (maison, locaux commerciaux, sites de production, etc.) en installant une ou plusieurs webcams reliées à un ordinateur. Ce dernier hébergera un serveur de vidéosurveillance permettant un mode de visualisation en ligne de ces locaux ainsi qu'une possibilité d'enregistrement de ces vidéos et en particulier de celles correspondant à des intrusions.

Vu la complexité de ce système, nous avons choisi de mettre en pratique les grandes principes de la méthodologie de développement RUP (Rational Unified Process) qui consiste à réaliser un développement itératif par l'analyse, la conception et le développement de plusieurs prototypes par enrichissement successif de leurs fonctionnalités. Ainsi, les grandes étapes de développement d'un tel système ont été identifiées comme suit :

- Développement d'un système de vidéo surveillance locale. Il s'agit de réaliser l'analyse, la conception et l'implémentation d'un premier prototype offrant les fonctionnalités de vidéosurveillance d'un local depuis la machine reliée aux webcams et qui constitue le serveur des vidéos.

- Développement, par enrichissement du système précédent, d'un deuxième prototype offrant les fonctionnalités de vidéosurveillance via le web. Il s'agit de permettre à un utilisateur d'accéder (visualiser) à distance, via une machine reliée au réseau internet, aux vidéos captées par une machine jouant le rôle du serveur.

- Développement du module « Alertes et gestion des intrusions ». Une fois le prototype offrant les fonctionnalités de visualisation en ligne d'un lieu distant est réalisé, le troisième objectif est le développement des fonctionnalités permettant à un utilisateur du système d'être alerté en cas de détection d'intrusion. Différentes possibilités de notification d'intrusion sont ainsi proposées.

- Développement du module « vidéosurveillance mobile ». IL s'agit de prendre en compte la mobilité de l'utilisateur et la possibilité qu'il soit alerté en cas d'intrusion.

L'étude du contexte de ce projet ainsi que sa réalisation sont détaillés dans la suite de ce document. Dans la section 2, nous présentons notre étude bibliographique centrée autour de la vidéosurveillance. Nous présentons dans cette section la notion de surveillance dans son sens général et on étudiera ensuite la vidéosurveillance, sa définition, ses buts et ses différents types en exposant quelques exemples de systèmes existants. Dans la section 3, nous présentons le système Rapace. Nous commençons cette section par la présentation de quelques éléments concernant la gestion de ce projet (rappel des objectifs, planification et répartition des tâches). Ensuite, nous détaillons les éléments liés à l'analyse et à la conception du système avant de présenter les différents aspects qui concernent son implémentation. La section 3.5 est consacrée à la présentation du prototype développé. Les différentes perspectives d'améliorations du système Rapace sont présentées dans la section 3.6. Nous terminons ce document par les sections présentons respectivement nos acquis et impressions personnelles, notre conclusion par rapport au travail réalisé et enfin un annexe regroupant un ensemble d'informations en relation avec le sujet de notre étude.

2 ETUDE BIBLIOGRAPHIQUE

2.1 INTRODUCTION : LA TELESURVEILLANCE

La télésurveillance est la surveillance à distance d'un lieu, public ou privé, de machines ou d'individus. Elle est employée dans de nombreuses situations, généralement pour des raisons de sécurité :

- Dans le cadre de la sécurité routière, au moyen de caméras spécialisées ou des capteurs à proximité voire même noyés dans la chaussée permettent d'évaluer la densité du trafic, les ralentissements qui peuvent en découler, la présence de personnes sur les bandes d'arrêt d'urgence, etc.
- Pour la surveillance des machines : divers capteurs permettent d'évaluer l'état de la machine, ces informations peuvent alors être envoyées à un poste de surveillance. L'épuisement de consommables, une anomalie de fonctionnement ou même un acte de malveillance serait alors détecté à distance ;
- Dans le cadre de la prévention de la délinquance (avec notamment la vidéosurveillance) ;
- Pour la surveillance de lieux sensibles (banques, centrales nucléaires, etc.) et d'habitations, afin de prévenir les intrusions, les cambriolages et les actes de vandalisme ;
- Dans le cadre de la télémédecine, et en particulier pour la surveillance des patients à distance ;
- Pour la surveillance à distance des enfants et des personnes vulnérables

2.2 LA VIDEOSURVEILLANCE

2.2.1 Présentation

La vidéosurveillance consiste à placer des caméras de surveillance dans un lieu public ou privé pour visualiser et/ou enregistrer en un endroit centralisé tous les flux de personnes au sein d'un lieu ouvert au public pour surveiller les allées et venues, prévenir les vols, agressions, fraudes et gérer les incidents et mouvements de foule.

Au début des années 2000, les caméras font leur apparition en nombre important dans de nombreuses villes européennes. Londres est réputée comme étant la ville où la vidéosurveillance est la plus importante. L'utilisation de la vidéosurveillance fait débat en matière de sécurité et de respect de la vie privée.

2.2.2 Son apparition

La vidéosurveillance s'est développée d'abord au Royaume-Uni, en réponse aux attaques de l'IRA(Armée républicaine irlandaise en anglais Irish Republican Army). Les premières expériences au Royaume-Uni dans les années 1970 et 1980 ont conduit à des programmes de grande ampleur au début des années 1990. Ces succès conduisirent le gouvernement à faire une campagne auprès de la population, et lança une série d'installations de caméras. Aujourd'hui, les caméras au Royaume-Uni couvrent la plupart des centres villes, et de nombreuses gares et parkings. Une étude donna le chiffre approximatif de 400 000 caméras à Londres et 4 millions au Royaume-Uni au total.

D'autres pays comme la France ont installé des systèmes de vidéosurveillance. En 1998 le nombre de caméras en France était estimé à un million dont 150 000 dans le domaine public. Ces caméras sont présentes dans divers lieux tels que les aéroports, les gares, les routes, les transports publics. Ces installations vidéo commencent aussi à fleurir dans les villes. À Avignon par exemple, une enquête à propos de la vidéosurveillance a révélé que 71 % des Avignonnais sondés étaient favorables à l'installation d'un tel système dans les parkings.

Cependant il existe aussi des associations qui militent contre toute forme de surveillance. C'est le cas de « Souriez, vous êtes filmés », pour n'en citer qu'une.

2.2.3 Ses buts

Les raisons de l'installation de systèmes de vidéosurveillance sont diverses, toutefois la sécurité publique ainsi que la protection des biens mobiliers ou immobiliers font office d'éléments phares dans la justification de la vidéosurveillance. En Angleterre, les attentats de juillet 2005 sont également un moteur pour l'augmentation du nombre de caméras.

Cette menace qui a toujours été présente n'a jamais vraiment créé un sentiment d'insécurité, mais les attentats du 11 septembre 2001 ont changé la donnée. Les gens ont pris conscience que personne n'était intouchable. Toutefois la mise en place de la vidéosurveillance ne peut s'expliquer uniquement par l'insécurité grandissante ou la protection des biens. Certaines autres raisons moins connues du grand public existent également. La mise en place de la vidéosurveillance permet une amélioration de la gestion des incidents ainsi qu'une augmentation de l'efficacité et de la rapidité d'intervention. Par exemple, dans la prévention du suicide ou encore lors d'accidents qui pourraient survenir sur la voie publique. Elle permet ainsi indirectement, de maintenir les primes d'assurances à un niveau raisonnable. La surveillance des axes routiers sert à informer en temps réel les automobilistes sur les conditions du trafic.

Quelques affaires de crimes ont été résolues grâce aux enregistrements fournis par les caméras de surveillance. Par exemple, après les attentats du métro de Londres du 7 juillet 2005, les enregistrements des caméras de surveillance ont été utilisés pour identifier les poseurs de bombes, bien qu'il soit admis qu'ils n'aient pas été indispensables. La question de savoir si la vidéosurveillance prémunit ou réduit les crimes n'a pas pu être montrée par les études indépendantes qui furent conduites que ce soit en France ou à l'étranger.
Le gouvernement britannique a jugé de son côté que les effets bénéfiques n'étaient pas possibles à évaluer, bien que Scotland Yard ait affirmé, en 2008, que la vidéosurveillance à Londres, qui compte 500 000 caméras, n'avait permis d'élucider que 3 % des vols dans la rue.

2.2.4 Domaines d'applications

La tendance de l'économie mondiale actuelle exige aux entreprises d'être réactive devant les demandes de plus en plus gourmandes de moyen de connectivite et infrastructure de communication et de marketing, la technologie de la vidéo sur réseau IP redynamise les applications de vidéosurveillance par de nombreuse fonctionnalité comme le contrôle a distance, la vidéo en temps réel. Ce qui rend plusieurs secteurs d'activités interactives.
On dénombre trois grandes catégories publiques dans lesquelles l'on retrouve ces systèmes de surveillance :

- Les aéroports, les transports publics et les gares.
- Les lieux publics et les parkings. Qui se verront principalement doter de systèmes classiques pour la surveillance globale bien que les aéroports commencent à adopter les mesures biométriques.
- Le trafic autoroutier. Qui pour sa part privilégiera les caméras qui ont la possibilité de reconnaître les véhicules.

Les installations privées importantes concernent les casinos et autres salles de jeux qui font régulièrement appel à des systèmes d'identification faciale pour reconnaître les fraudeurs.

2.2.5 Evolution de la vidéosurveillance

Les premières caméras avaient des images de basse qualité et noir et blanc, sans possibilité de zoomer, ni de changer l'angle de vue. Les caméras modernes les plus performantes sont en couleur, permettent des zooms et une mise au point très nette. Les dispositifs d'enregistrement et d'analyse sont plus précis, plus efficaces.

La loi de la République Française en vigueur définit dans l'arrêté du 3 août 2007 (publié au Journal officiel le 21 août, avec son rectificatif du 25 août), les normes techniques des images. A de rares exceptions près, la définition requise est dite 4 CIF, soit 704 x 576 pixels.

Définition très rarement atteintes par les anciennes caméras ou même certaines toujours sur le marché, en général en CIF soit 352 x 288 pixels, ou VGA, soit 640 x 480 pixels. Il est possible d'avoir une caméra de résolution plus faible si elle permet de prendre une "vignette de visage" pour identification de 90x60 pixels. Sur les anciennes caméras, cela signifie que le visage doit représenter 5% environ de la superficie de l'image (1% en 4 CIF). Par ailleurs, le nombre d'images par seconde requis est de 6 ou 12, selon la situation, lente ou rapide, à surveiller. Les nouvelles installations doivent évidemment se conformer à la loi, tandis que les anciennes ont jusqu'au 21 août 2009 pour se mettre en conformité. C'est l'utilisateur du système qui est responsable de sa conformité à la loi.

En pilotant ces caméras avec des ordinateurs, il est possible de suivre des mouvements, il est par exemple possible de déceler des mouvements dans un endroit où il ne devrait pas y en avoir, ou au contraire se focaliser sur un individu et le suivre à travers la scène. L'informatique peut faire coopérer plusieurs caméras pour le suivre dans un espace urbain entier.

L'une des évolutions les plus probables de la vidéosurveillance est le rapprochement des enregistrements avec des données biométriques. Cette technologie permettrait par exemple aux ordinateurs d'analyser la démarche des passants : une personne lourdement chargée adopte une démarche inhabituelle ; que transporte-t-il ? Des explosifs, des armes, une caméra de télévision ou des bouteilles de soda ? De même, des recherches récentes misent sur la prévisibilité du comportement humain dans les espaces publics : un voleur ne se comporterait pas de la même façon qu'un usager. L'ordinateur peut identifier ce genre de mouvements et donner l'alerte.

Couplées à une base de données biométrique, il devient possible de déterminer l'identité d'une personne sans l'aborder et sans même qu'elle ne s'en rende compte. Une expérience de ce type eut lieu en 2007 dans une gare à Mayence, en Allemagne ; 60% des volontaires furent identifiés parmi une foule de 20 000 personnes. Ce résultat est trop faible pour une mise en application mais ces promoteurs pensent pouvoir proposer une technologie convaincante d'ici 2012.

2.2.6 LES DIFFERENTS TYPES DE SYSTEMES

2.2.6.1 Système sur réseaux IP

Ce système relie un réseau de caméras IP, qui peut compter de nombreuses unités, à un système d'enregistrement numérique. D'une part, cela permet de pouvoir stocker une quantité importante d'images, sans perte de qualité, tout en pouvant les consulter rapidement grâce à des logiciels de traitement. D'autre part, le fait d'informatiser un système de surveillance permet de profiter des technologies de communication comme Internet. Ainsi, les caméras sont « visibles » et gérables depuis n'importe où dans le monde. L'évolution des téléphones mobiles a créé la "vidéosurveillance mobile" avec l'accès aux vidéos via Internet mobile sur PDA ou via GSM GPRS sur téléphone GSM doté de Java. Cette technologie permet également d'économiser et de mutualiser les câbles réseaux qui sont généralement disponibles dans les bâtiments récents.

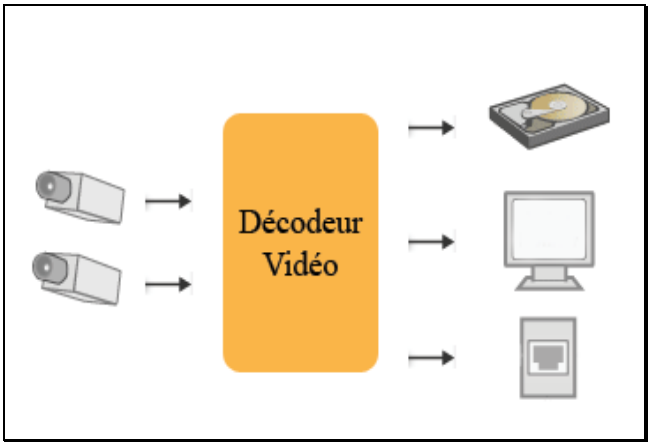


Figure 1. Vidéosurveillance sur réseau IP

2.2.6.2 Kit de vidéosurveillance

On entend par « kit » le genre de caméras utilisées dans les petits magasins, par exemple. Il regroupe en général une ou deux caméras et un moniteur.

Ces systèmes sont plutôt utilisés à titre de prévention et n'enregistrent pas ce qu'ils voient. C'est en quelque sorte de la vidéosurveillance bon marché qui est proposée comme une solution de sécurité peu coûteuse.

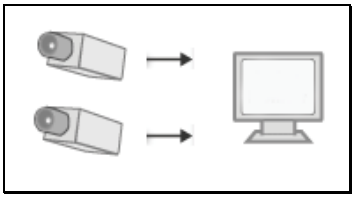


Figure 2. Kit de vidéosurveillance

2.2.6.3 Réseau « classique » de vidéosurveillance

Le réseau est basé sur un système analogique, avec dans la plupart des cas un enregistrement limité dans la durée. Il s'agit là d'une des méthodes les plus anciennes donc également des plus répandues dans un grand nombre d'établissement. Cependant, ces systèmes ne répondent plus, à de très rares exceptions près, aux nouvelles exigences techniques de l'arrêté du 3 août 2007.

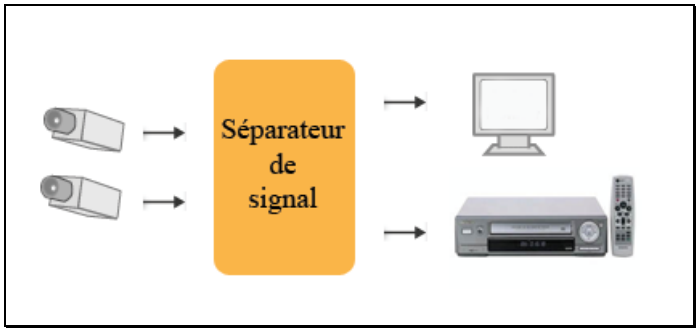


Figure 3. *Système de vidéosurveillance analogique*

2.2.6.4 Système « hybride » de vidéosurveillance

Les systèmes hybrides intègrent les systèmes classiques de vidéosurveillance basés sur les caméras analogiques et les caméras en réseau. Il permet d'intégrer aisément les deux types de systèmes en place sur un seul serveur ou de faciliter l'évolution d'un système de vidéosurveillance analogique vers le numérique, sans remettre en cause l'existant, et introduire de nouvelles fonctions comme la détection de disparition / apparition d'objet et le comptage d'objets ou de personnes.

2.2.7 EXEMPLES DE QUELQUES SYSTEMES EXISTANTS

2.2.7.1 Le système ASCAM-2E2I

La solution ASCAM intègre un logiciel de vidéo surveillance sur réseau IP, un logiciel d'enregistrement sur mouvement avec les fonctions magnétoscope et finalement des caméras réseau IP jour / nuit pour extérieur en caisson étanche et pour intérieur.

L'accès aux caméras et aux enregistrements sur alarme par Internet permet d'assurer une surveillance à distance. Les logiciels permettent un monitoring temps réel sur site et à distance via Internet ainsi que l'enregistrement sur détection de mouvement, planning et alarme

La détection de mouvement peut être configurée selon les critères suivants :

- Champs de vision des caméras de surveillance
- Zones de détections, masques de détection
- Objet manquant (détection de vol)
- Nouvel objet statique (colis douteux)
- Focus déréglé, caméra obstruée (vandalisme)

Enfin la notification d'alarmes comprend l'envoi d'email d'alerte, appel téléphonique et également une alarme sonore

2.2.7.2 Le système ASCAM-NUUO

La solution ASCAM-NUUO intègre un serveur de vidéo surveillance permettant une visualisation à distance via internet sur PC et sur téléphone mobile 3GPP. Elle fournit un ensemble de fonctionnalités que l'on peut résumer dans la liste suivantes :

- Visualisation des flux vidéo en local sur moniteur
- Alarme et visualisation en levée de doute, fenêtre pop up
- Console d'administration des paramètres de gestion
- Gestion des droits d'accès: profils, horaires, à distance
- Alarme sur détection de mouvement
- Enregistrement possible sur Planning (Schedule)
- Etc.

3 LE SYSTEME RAPACE

3.1 GESTION DU PROJET

3.1.1 Synthèse du cahier des charges

L'objectif de ce TER que nous avons énoncé lors de l'élaboration du cahier des charges est de réaliser un système qui permet à une personne de surveiller un local. Pour cela, le système nécessite seulement l'installation de caméras/webcams reliées à un ordinateur.

Afin de mettre en œuvre ce système de vidéosurveillance, nous nous sommes fixés plusieurs sous-objectifs (vidéosurveillance locale, vidéosurveillance à distance…) qui constituent des étapes en vue de l'aboutissement du système.

Concernant l'architecture du système, notre projet rentre dans le cadre d'une application client/serveur. Le serveur aura à sa charge l'attente de la connexion de l'utilisateur, l'attente des connexions dédiées pour les différentes webcams, la réception et l'analyse des messages émanant du client. De plus, en cas d'intrusion, le serveur exécutera les actions choisies par le client.

Le client, quant à lui, s'occupera de la connexion au serveur, de l'envoi de messages, la réception et l'affichage des flux capturés. De plus, il pourra choisir les actions à effectués en cas d'intrusion (envoi de SMS, d'email…).

3.1.2 Planification et répartition des tâches

Dès le début du projet nous avons voulu suivre une démarche visant à structurer, assurer et optimiser le bon déroulement du projet pour être planifié dans le temps et surtout pour atteindre le niveau de qualité souhaité dans le meilleur délai possible. Ainsi, dans un premier temps on a déterminé, de façon claire, les objectifs du cahier des charges. Ensuite on les a numérotés et datés selon leur évaluation tout en réservant un certain temps pour se rattraper en cas d'éventuels problèmes.

Le résultat de cette étape est présenté dans le diagramme de Gantt de la figure 4 ci-dessous:

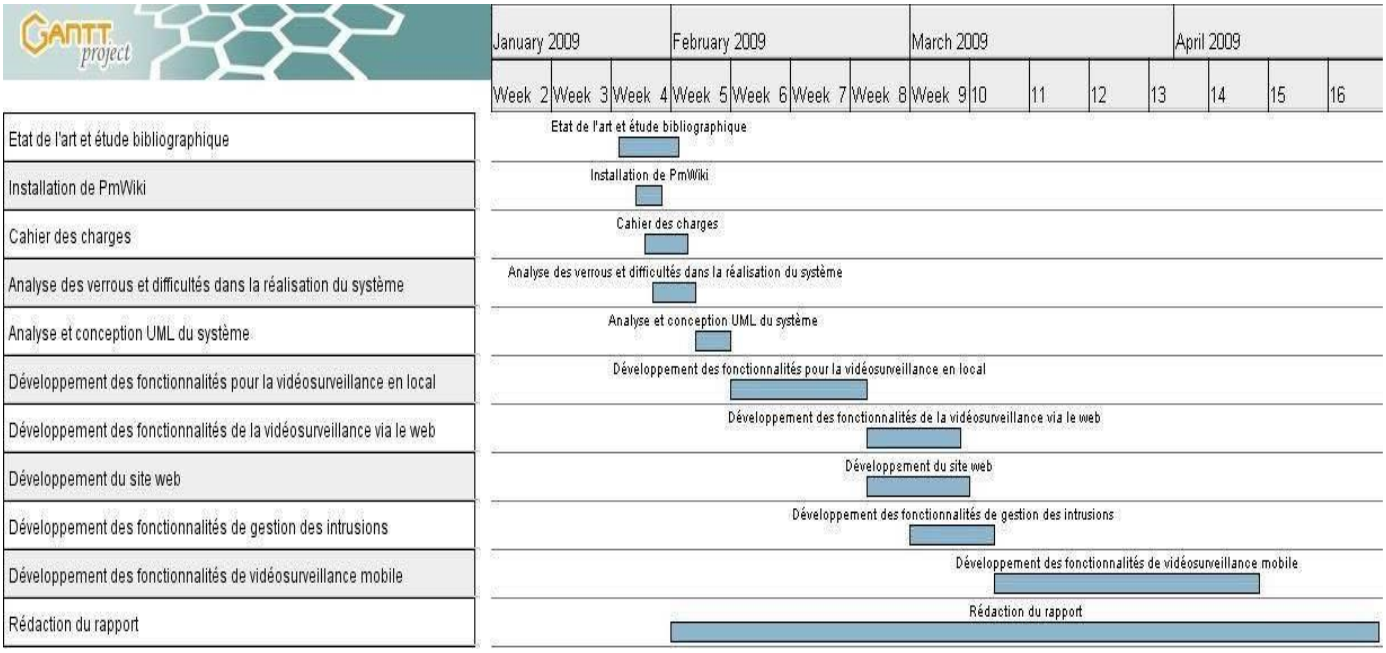


Figure 4. *Diagramme de Gantt*

Nous avons ensuite réparti les différentes tâches identifiées entre les membres du groupe. Nous présentons ci-dessous que la répartition réelle des tâches résultat de différents changements dus à la prise en compte de certains paramètres(tels que le non disponibilité d'une des membres du groupe).

Membres / Taches	Abderrarmane Seriai	Mohammed Ferjani	Annes Seddik	Hakim Hennani
Etat de l'art et étude bibliographique	***	**	*	**
Installation du Wiki collaboratif	***	*		**
Cahier des charges	***	***	*	***
Analyse des verrous et difficultés dans la réalisation du système	***	***	*	***
Analyse et conception UML du système	***	***		**
Développement des fonctionnalités de la vidéosurveillance en local	**	***		
Développement des fonctionnalités de la vidéosurveillance via le web	***	***		**
Développement du site web	**	**		***
Développement des fonctionnalités de gestion des intrusions	*	***		
Rédaction du rapport	***	***		***

Légende :
* : Contribution.
** : Contribution considérable.
*** : Responsable tâche.

3.2 ANALYSE ET CONCEPTION DE RAPACE

3.2.1 Analyse et étude des fonctionnalités

L'objectif de cette étape primordiale est de créer une représentation simplifiée du problème 'le modèle' et de sa solution. Le modèle constitue ainsi une représentation possible du système pour un point de vue donné : le notre dans le présent rapport.

3.2.1.1 Les fonctions offertes

Le diagramme de cas d'utilisation (use case) de la figure 5 ci-dessous, met en évidence les grandes relations fonctionnelles entre les acteurs (principaux) et le système.

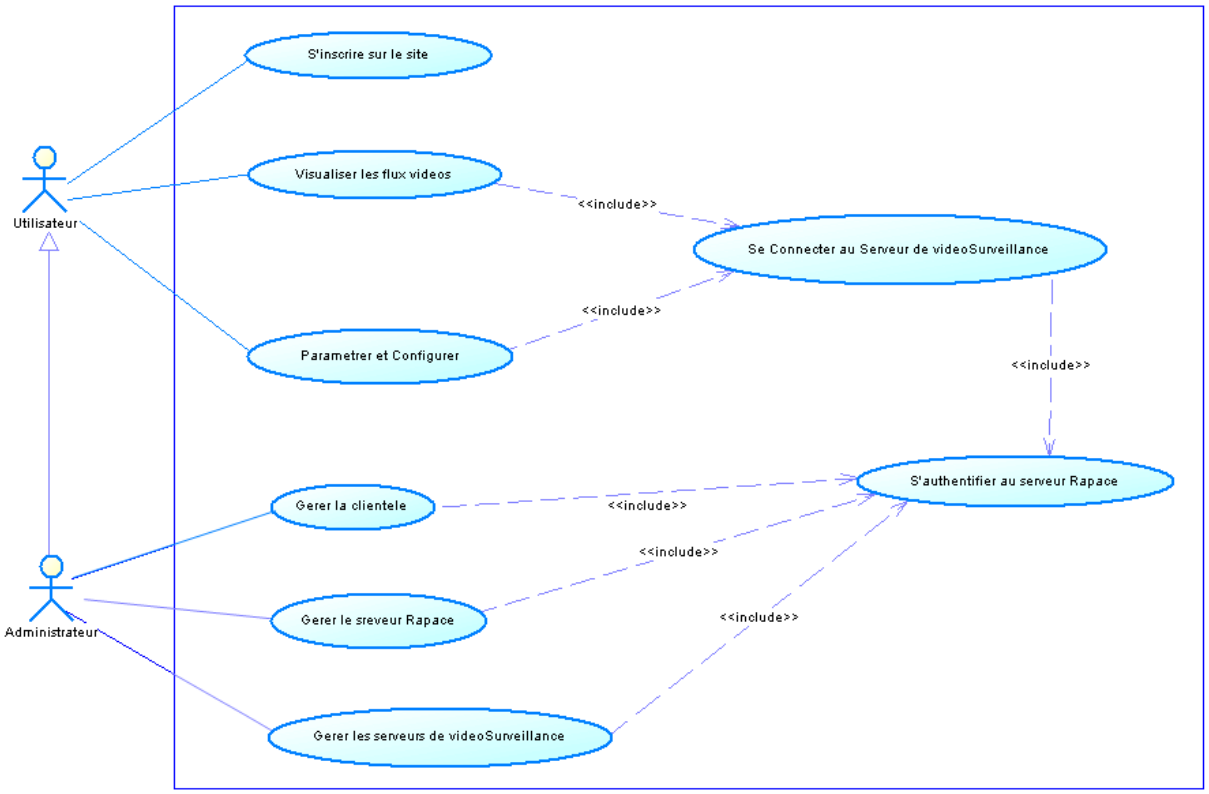


Figure 5. *Diagrammde de cas d'utilisation*

Remarque : Les fonctionnalités de visualisation et de paramétrage et configuration sont les plus importantes dans notre système c'est pour cela qu'on va les détailler dans la suite.

3.2.1.2 Etude détaillée de la fonction de visualisation

Cette fonctionnalité se décompose en deux sous-fonctionnalités :

- La visualisation des flux en temps réel, c.à.d. ceux qui transitent sur les webcams au moment de la visualisation.
- La visualisation des vidéos qui ont été enregistrées en cas d'intrusion.

La figure 6 ci-dessous détaille cette fonctionnalité.

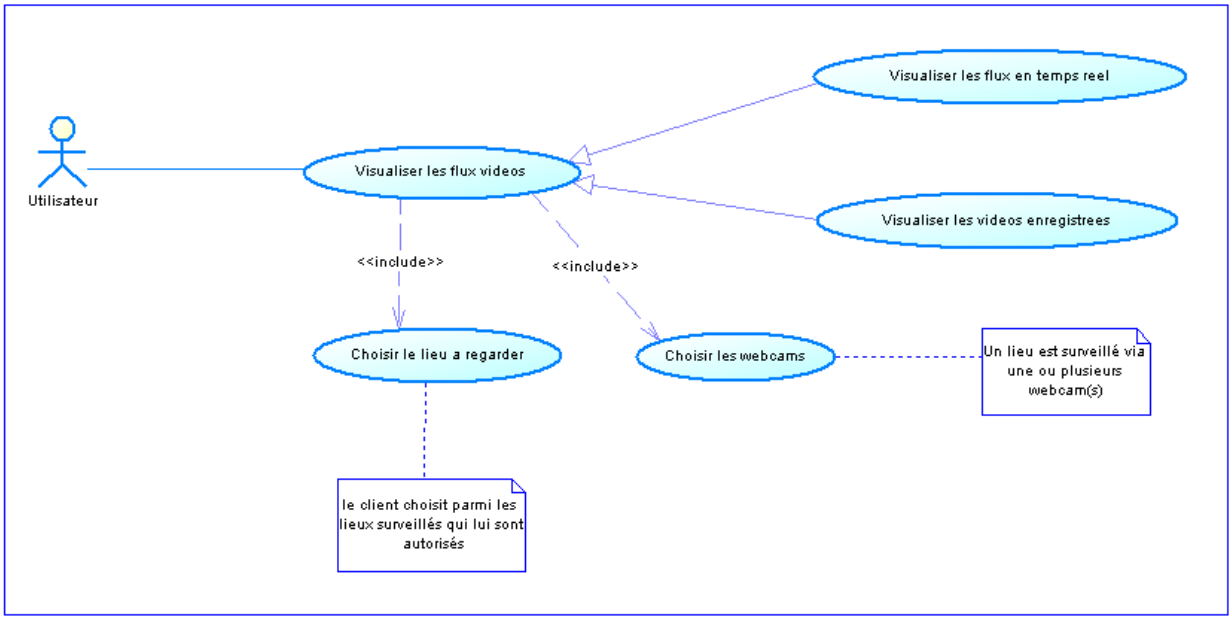


Figure 6. *Detaille de la fonctionnalité de visualisation*

3.2.1.3 Etude détaillée de la fonction de paramétrage et configuration

La figure 7 ci-dessous montre les différentes possibilités de configurations à disposition de l'utilisateur :

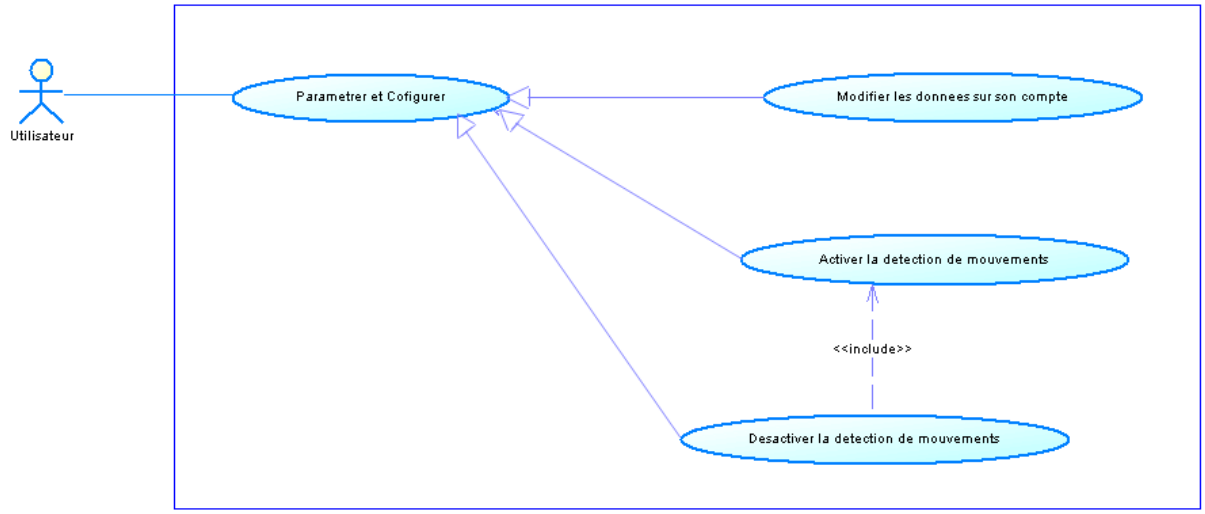


Figure 7. *Detaille de la fonctionnalité de paramétrage et configuration*

La fonctionnalité d'activation de la détection de mouvement incorpore d'autres sous-fonctionnalités comme le montre la figure 8 ci-dessous :

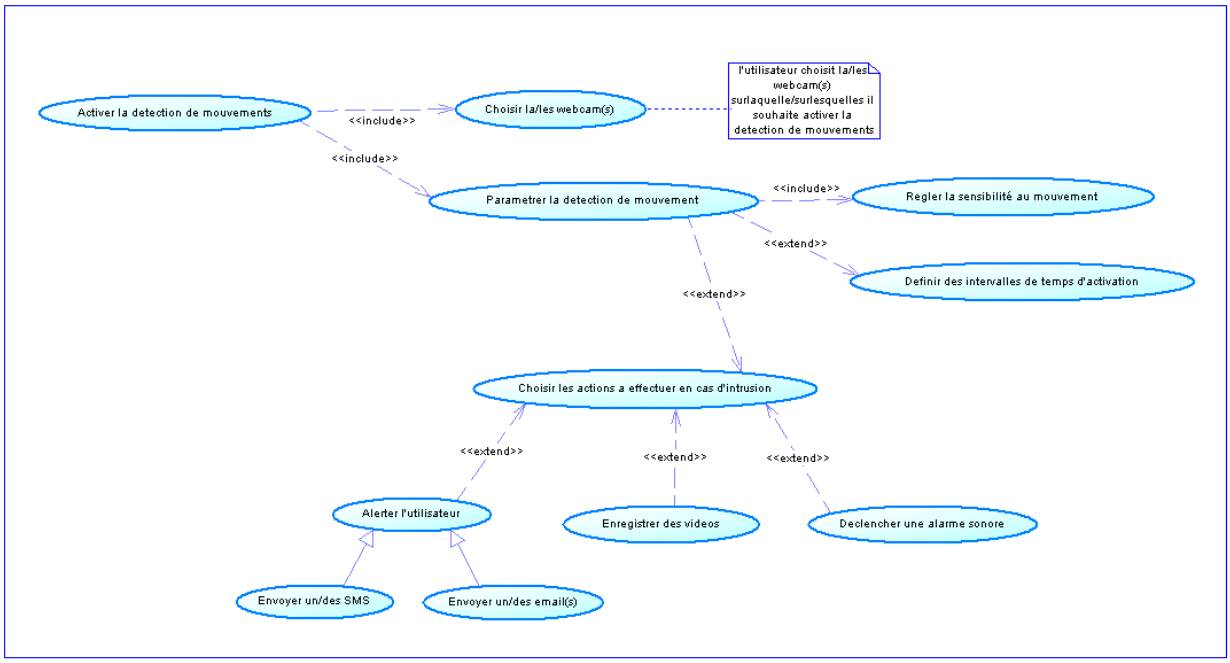


Figure 8. *Detaille de la fonctionnalité d'activation de la détection de mouvement*

3.2.2 Etude architecturale

L'étude conceptuelle que nous venons de présenté ci-dessus nous a permis d'effectuer un constat par rapport à la suite de la conception de notre système. Nous nous sommes aperçus qu'il était nécessaire de bien dissocier deux taches bien distinctes au sein de notre application : une tache qui sera dédié à la gestion des différentes webcams et une autre dédié à l'envoi des différents flux récupérés.

Cependant, ce redécoupage des taches nous a confronté à la problématique suivante : faut-il cohabiter ces deux taches dans un seul serveur ou bien les séparer dans un serveur différent.

Afin de résoudre ce problème, il a été nécessaire de réaliser une étude architecturale que nous allons vous présentées ci-dessous.

3.2.2.1 Architecture multiserveurs

3.2.2.1.1 Fonctionnement

La première architecture pouvant matérialiser le fonctionnement du système désiré consiste en la coexistence de plusieurs serveurs dont chacun est relié d'un coté à des webcams et d'autre coté à une connexion internet. De cette manière, le client lorsqu'il se connecte au serveur, pourra visualiser directement le flux vidéo récupéré sans être redirigé (comme le montre la figure 9),

Par conséquent, chacun de ces serveurs sera amené à effectuer les traitements qu'il doit gérer (attente de la connexion du client, réception et analyse des messages émanant du client…).

Le client s'occupera de la connexion au serveur (chaque client possède sa propre adresse http qui le relie à son serveur local), de l'envoi de message vers le serveur (réception et affichage des flux capturés (par le serveur) sur les différentes webcams, choix des actions à exécuter en cas d'intrusion.

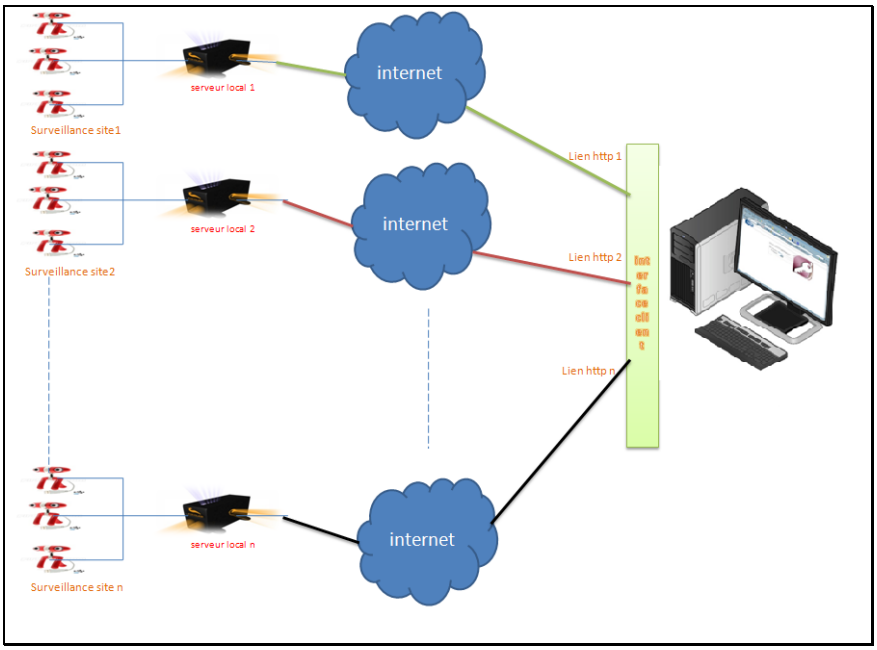


Figure 9. *Schéma de l'architecture multi serveurs*

3.2.2.2 Avantages et inconvénients

La liaison directe entre le serveur et le client (c'est-à-dire il n'y aura pas de redirection et de renvoi de flux récupère par le serveur vers un autre serveur et après vers le client) ainsi que le fait que chaque client a sa propre adresse http font de cette architecture une solution convenable.

Néanmoins, elle présente pas mal d'inconvénients. Par exemple, si on modifie l'interface chez l'un des clients, on sera obligé d'appliquer cette modification autant de fois qu'au nombre des clients. De plus, tout le système dont le serveur http, notre application … est installée chez le client. Enfin, il n'y a pas d'administration.

3.2.2.3 Architecture avec forte centralisation

3.2.2.3.1 Fonctionnement :

Dans cette architecture nous avons rajouté un serveur central qui sera relié d'une part à des serveurs locaux et d'autre part à une connexion internet, comme ca le client quand il demande la visualisation de flux de vidéo, ce flux va être envoyé d'abord du serveur local vers le serveur central et puis de ce dernier vers le client, donc le serveur central il joue le rôle d'intermédiaire entre le client et le serveur local et il permet aussi l'enregistrement des séquences de vidéo a son niveau après la demande du client.

Les traitements à effectuer au niveau du serveur local sont les mêmes que dans l'architecture précédente.

Et pour le client Les traitements sont les même que dans l'autre architecture sauf que dans cette architecture il y'aura une seule adresse HTTP pour tous les clients.

3.2.2.3.2 Avantages et inconvénients:

Les avantages de cette architecture sont qu'il n'y a qu'une seule adresse http pour tous les clients. Le client peut visualiser des vidéos qui sont enregistrée sur le serveur central même si le serveur local correspondant tombe en panne. De plus, l'administrateur peut de virer ou d'ajouter des clients car ce n'est qu'à partir de ce serveur que le client peut visualiser les vidéos ; chose qui n'était pas possible dans l'architecture précédente.

Par contre, cette architecture est très lourde car les vidéos sont envoyées de serveur local vers le serveur central puis elles sont visualisées. Autre inconvénient, si le serveur central tombe en panne on ne peut pas visualiser les vidéos bien que le serveur local soit capable de récupérer les flux des vidéos.

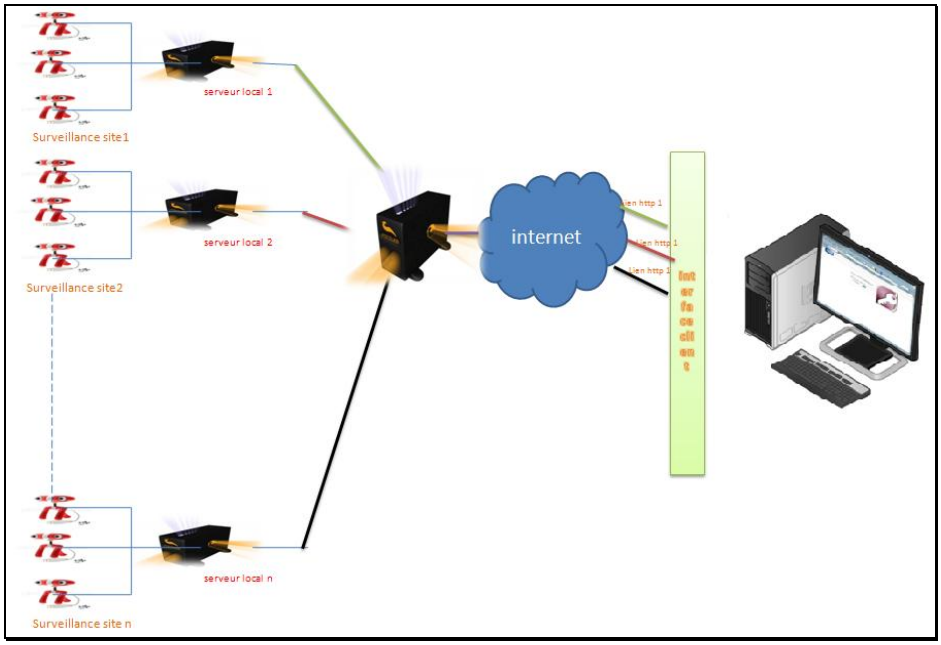


Figure 10.*Schéma de l'architecture avec forte centralisation*

3.2.2.4 Architecture avec centralisation souple

3.2.2.4.1 Fonctionnement :

Cette architecture sera presque la même que la précédente sauf que dans celle-ci on a essayé d'enlever le principal inconvénient de la précédente (la lourdeur) en mettant le client en communication directe avec le serveur local (via l'ouverture d'une session de travail) après l'obtention de l'ok du serveur central bien sur et comme ca on aura plus des flux qui seront retransmis ou redirigés (comme le montre la figure 11)

Dans cette architecture, le serveur central sera donc relié d'une part à des serveurs locaux et d'autre part à une connexion internet, et quand le client demande la visualisation de flux de vidéo récupéré par le serveur local, le serveur central ouvre une session entre ce dernier et le client et

comme ca le client sera en communication directe avec le serveur local et du cout le flux va être envoyé une seule fois

Les traitements à effectuer au niveau du serveur local et du client sont les mêmes que dans l'architecture précédente.

3.2.2.4.2 Avantages et inconvénients :

Cette architecture présente une multitude d'avantages. Dans un premier temps, elle permet une liaison directe entre le serveur et le client : c'est-à-dire il n'y aura pas de redirection et de renvoi de flux récupère par le serveur. Dans un second temps, l'administrateur a la possibilité de supprimer ou d'ajouter des clients car ce n'est qu'a partir de ce serveur que le client peut visualiser les vidéos, ce qui n'était pas possible dans l'architecture précédente.

De plus, il n'y a qu'une seule adresse http pour tous les clients. Enfin, le client à la possibilité de visualiser des vidéos qui sont enregistrées sur le serveur central même si le serveur local correspondant tombe en panne.

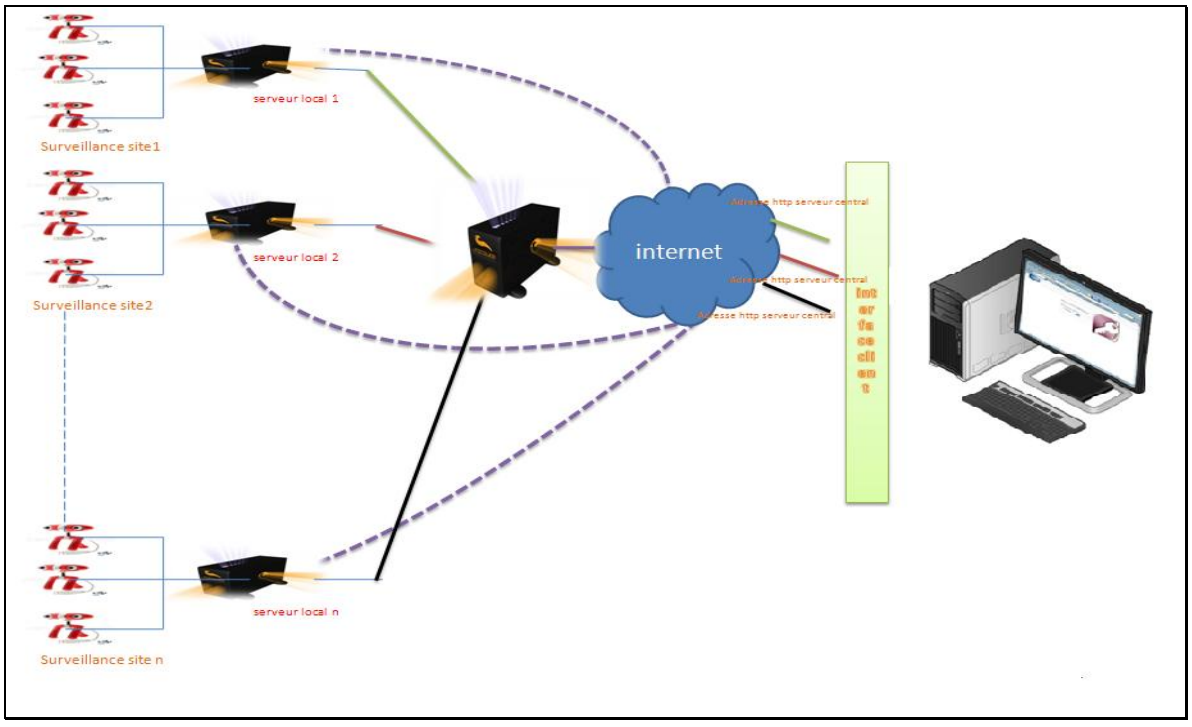


Figure 11.*Schéma de l'architecture avec centralisation souple*

3.2.2.5 *Modélisation de l'architecture retenue*

3.2.2.5.1 Structure et composants de l'architecture

La modélisation objet n'est pas un processus linéaire mais plutôt itératif. En d'autres mots, on est sensé itérer et raffiner à chaque étape du cycle de vie du projet.

Voici donc la première version 'le modèle de conception' du diagramme de classes :

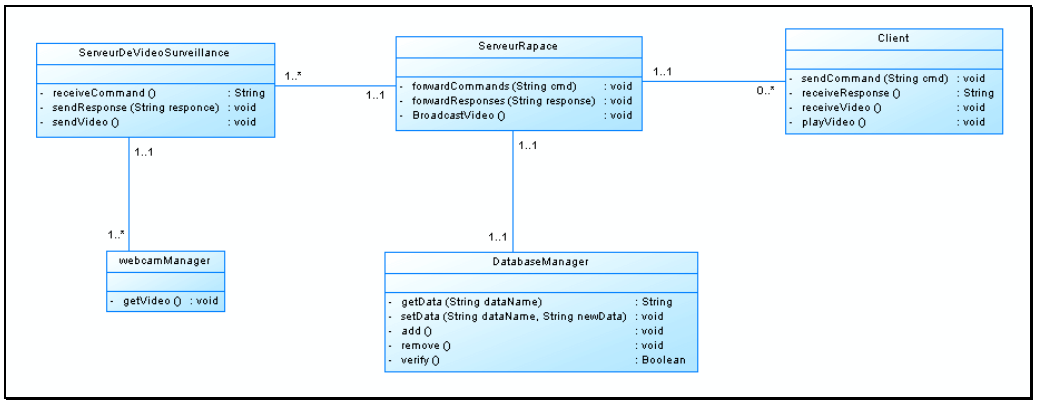


Figure 12.*Diagramme de Classes : modèle de conception*

Après les raffinages appliqués sur le diagramme de classes au fur et à mesure qu'on avançait, on a aboutit au diagramme d'implémentation que montre la figure 13.

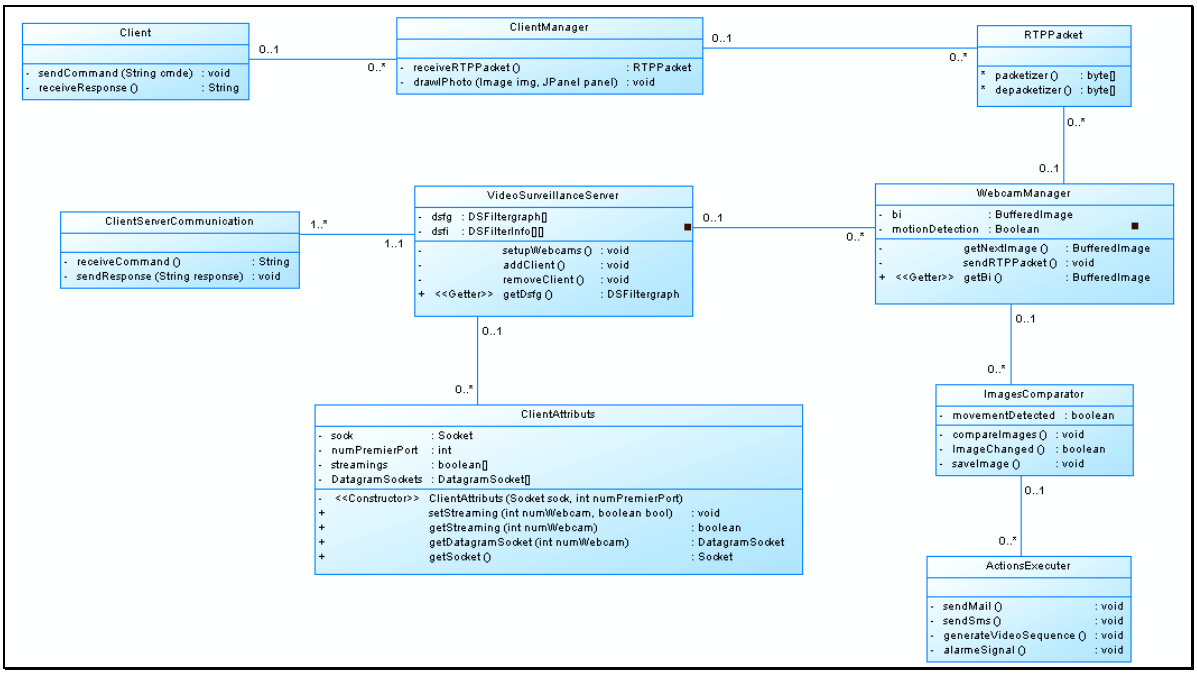


Figure 13.*Diagramme de classes : modèle d'implémentation*

3.2.2.5.2 Comportement et fonctionnement de l'architecture

Pour représenter les interactions entre objets à travers le temps, typiquement entre un utilisateur ou un acteur et les objets et composants avec lesquels ils interagissent au cours de l'exécution du cas d'utilisation, voici un diagramme de séquence représentant un 'scénario' du Cas d'Utilisation.

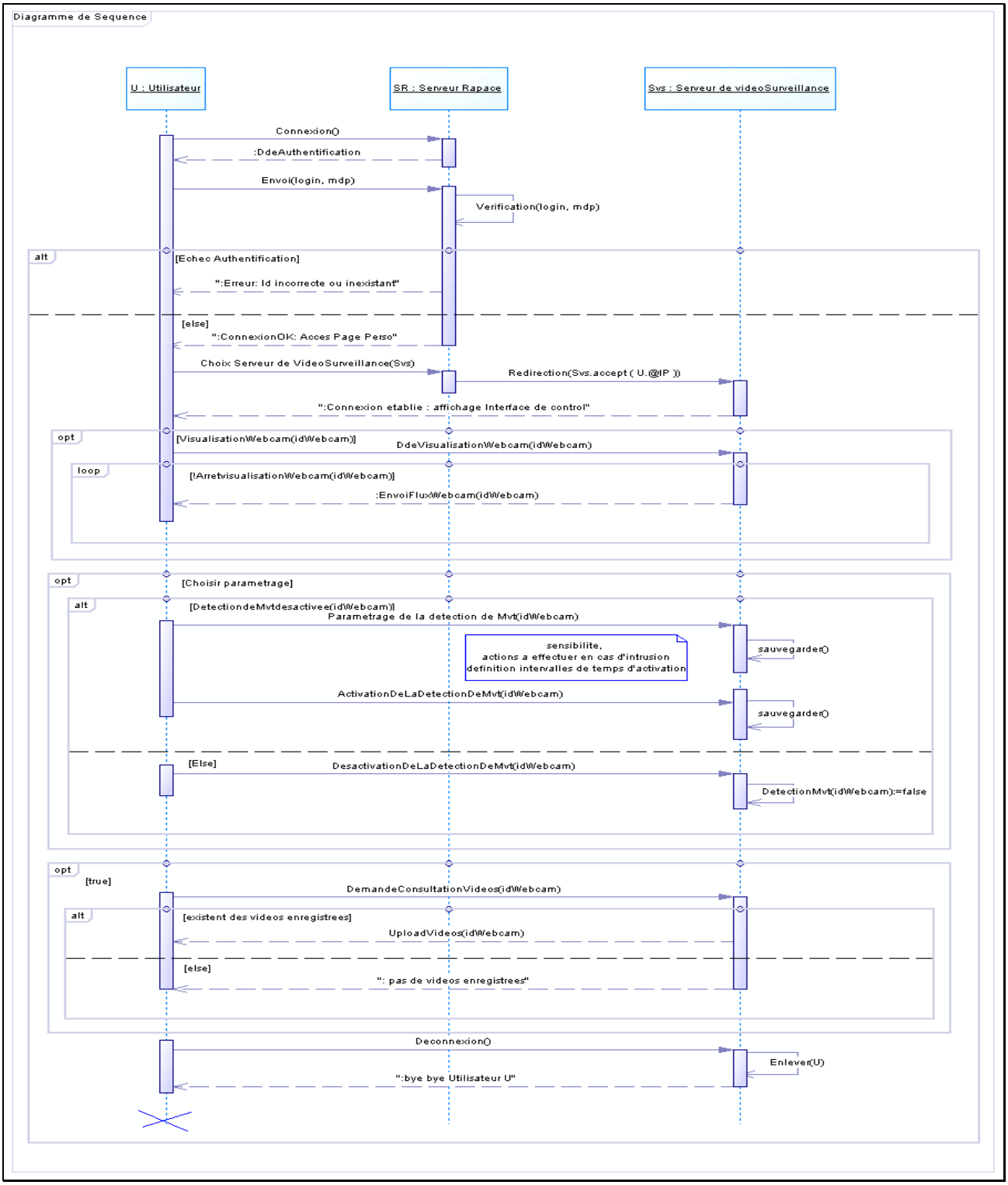


Figure 14.*Diagramme de séquence*

3.2.3 Analyse pré-implémentation

Après avoir fait l'étude conceptuelle et architecturale il était nécessaire de faire une étude concernant le choix de langage et les différents verrous qu'on peut rencontrer au cours de l'implémentation

3.2.3.1 Choix des langages

Le choix de langage représente une étape très importante dans la réalisation de n'import quelle application parce c'est à partir de cette étape qu'on fait la correspondance entre les solutions que nous offre le langage et les résultats souhaités Java est un langage orienté objet permettant de créer des applications aux fonctionnalités poussées et ce avec un minimum de lignes de codes.

Nous allons tirer profit de ces qualités pour détailler la conception d'un système de vidéo surveillance basé sur ce langage.

Nous l'avons choisi comme langage pour programmer notre projet, car d'un coté on est plus familiarisés avec java que d'autres langages et d'autre coté parce que avec ce langage on peut faire des applications de bureau comme on peut faire des applications web ou de réseau ce qui convient bien avec les buts de notre application. Concernant le développement web nous avons choisi d'utilisé toutes les technologies offerte par le web2.0 (Ajax, CSS, javascrip, php)

3.2.3.2 Analyse des verrous d'implémentation

Lors de notre analyse pré-implémentation de notre système, nous avons été confrontés à plusieurs verrous dans différents domaines.

Au niveau des pilotes (drivers) des périphériques matériels, il a fallu réfléchir à comment récupérer la liste des sources de données installées sur la machine, comment utiliser un périphérique externe (webcam) comme source d'entrée dans un programme.

La lecture d'un flux de données média nous a posé le problème de la connexion avec la webcam, de la gestion de plusieurs webcams.

Concernant la surveillance à distance, nous avons décelé deux verrous importants : la transmission d'un flux vidéo en temps réel sur internet et le codage à l'envoi et décodage à la réception.

A propos de la gestion d'intrusion, nous nous sommes demandés comment détecter un mouvement et que faire en cas d'intrusion. Enfin, nous avons étudié la question de l'envoi de mail ou de SMS.

3.3 IMPLEMENTATION DU FONCTIONNEMENT DE L'ARCHITECTURE

L'étude architecturale nous a permis d'identifié la meilleure architecture permettant de répondre au type de fonctionnement nécessaire pour notre système par conséquent nous avons opté pour le mécanisme de redirection comme moyen pour matérialiser le fonctionnement en question. Ce dernier consiste à faire basculer le client entre les deux serveurs (central et local) tout en gardant la main du serveur central sur le serveur local.

Différentes solutions sont possibles : utilisation simple du protocole http, la cohabitation du protocole http et la notion d'applet, utilisation de la notion de redirection directe et utilisation de la notion de redirection indirecte

3.3.1 Utilisation simple du protocole http

Le protocole HTTP (HyperText Transfer Protocol) est le protocole le plus utilisé sur Internet depuis 1990. La version 0.9 était uniquement destinée à transférer des données sur Internet (en particulier des pages Web écrites en HTML. La version 1.0 du protocole (la plus utilisée) permet désormais de transférer des messages avec des en-têtes décrivant le contenu du message.

Le but du protocole HTTP est de permettre un transfert de fichiers (essentiellement au format HTML) localisés grâce à une chaîne de caractères appelée URL entre un navigateur (le client) et un serveur Web.

Malgré toutes les solutions que le protocole http a pu apporter à notre application, son utilisation seule n'a pas répondu à notre but principal

3.3.2 La cohabitation du protocole http et de la technologie des applets

Dans cette approche on a essayé de s'appuyer sur la technologie d'applet en l'incorporant dans une page html dans le but de pouvoir utiliser du code java au sein d'une page html et du coup la possibilité d'utiliser la notion de socket pour relier les deux serveurs avec le client mais Les résultats obtenus n'ont été satisfaisants parce que on a découvert que sur le réseau une applet n'est pas autorisée à :

- Établir des connexions réseaux avec autre ordinateurs à l'exception de celui a partir duquel le code a été chargé
- Accepter des connexions réseaux d'un hôte autre que celui a partir duquel elle a été chargée

3.3.3 L'utilisation de la notion de redirection directe

Selon les études qu'on a fait et qui vont être présentées ci-dessous sur cette approche on a trouvé que ca convient partiellement avec notre but principal

3.3.3.1 Les différents types de redirection

Il existe plusieurs cas pour lesquels des redirections doivent être utilisées (renommage d'un fichier, changement d'adresse du site, lien…). Du point de vue du référencement, parmi toutes les méthodes de redirection, certaines sont plus recommandées que d'autres, comme nous l'expliquons ci-dessous. Du point de vue de l'internaute, c'est en général plus simple car il suffit de trouver un moyen d'afficher la page redirigée : il ne sera pas sensible au type de redirection.

Voici les techniques de redirection les plus courantes :

- Redirection directement sur le serveur
- Redirection par URL Rewriting
- Redirection dans un script serveur (PHP, ASP, et.
- Redirection par balise META Refresh
- Redirection JavaScript

3.3.3.2 Les avantages et les inconvénients

Le principe de redirection directe ne permet pas au serveur central de garder la main sur la communication entre le client et le serveur local. De plus, le client n'a pas besoin de passer par le serveur central pour accéder au serveur local après avoir le lien du serveur local

3.3.4 L'utilisation du principe de redirection indirecte

Tout en essayant d'éviter les inconvénients de l'approche précédente on est arrivé a cette solution adéquate à notre but principale

3.3.4.1 Fonctionnement :

Le client, quand il veut visualiser la vidéo, se connecte au serveur central en envoyant le login et le mot de passe, et attend la réponse du serveur

Lorsque le serveur reçoit la demande du client, il vérifie son identité en comparant les informations envoyées par ce dernier et celles enregistrées dans la base de données.

S'il trouve que les informations sont identiques il lui envoie l'ensemble des liens des serveurs locaux qui lui correspondent et ca après avoir envoyé ces informations aux serveurs locaux concernés.

Sinon il lui envoie une page d'erreur disant que les informations envoyé sont erronées

3.3.4.2 Avantages et inconvénients:

Les avantages de ce principe de redirection sont que le serveur central peut garder la main sur la communication entre le client et le serveur local. De plus, le client a toujours besoin de passer par le serveur central pour accéder au serveur local

Pour plus de détails voir l'Annexe6

3.4 REALISATION DES DIFFERENTS MODULES

Dans cette partie, nous allons vous exposer dans un premier temps le module d'identification et d'acquisition. Dans un second temps, nous aborderons le module de diffusion. Enfin, nous traiterons le module d'intrusion.

3.4.1 Réalisation du module identification et acquisition

3.4.1.1 Objectif

Arriver à identifier toutes les webcams branchées sur la machine et à acquérir les flux qui y transitent. Le travail avec le multimédia présente plusieurs chalenges. Les flux Multimédias contiennent une grande quantité de données qui doivent être traitées très rapidement. Audio et vidéo doivent être synchronisées d'une sorte qu'il démarre et s'arrête en même temps, et joue au même taux. Les données peuvent provenir de plusieurs sources, y compris les fichiers locaux, les réseaux informatiques, les émissions de télévision, et des caméras. De plus, ils proviennent d'une variété de formats, tels que Audio-Vidéo Interleaved (AVI), Advanced Streaming Format (ASF), Motion Picture Experts Group (MPEG), et de Digital Video (DV). Enfin, Le programmeur ne sait pas à l'avance quel matériel sera présent sur l'utilisateur final du système.

3.4.1.2 Etude des solutions possibles

A. JMF (Java Media Framework) :

JMF est une API Java permettant de manipuler aisément toutes sortes de contenus multimédia avec Java tels que du son ou de la vidéo. Elle offre les outils nécessaires pour faire de l'acquisition, du traitement et du transport de médias basés sur le temps. L'avantage est de pouvoir concevoir des applications utilisant des éléments multimédias (Webcam, micro, vidéos…) et pouvant s'exécuter sur différentes plates-formes logicielles (principalement Windows et Linux). La version actuelle de JMF est la 2.1.

Cette API est une initiative de SUN qui souhaite apporter une solution « time-based media processing » (traitement de media basé sur un timeline) à Java. Les média basé sur le temps sont des données qui changent par rapport au temps. Nous les retrouvons bien entendu dans les vidéos, l'audio, les séquences MIDI et autres animations.

Pour plus de détails, voir Annexe1.

Cependant, JMF ne fait pas de distinction entre les différentes webcams installées sur la machine, Il les prend toutes pour : « vfw:Microsoft WDM Image Capture (Win32):0 ». Du coup, on ne peut en utiliser qu'une seule à la fois, sans même pouvoir la spécifier.

Nous nous sommes alors orientés vers une deuxième solution, l'API FMJ Project.

B. FMJ Project (Freedom for Media in Java):

Projet Open-source dans le but de fournir une alternative à JMF. Comme il est compatible avec JMF, on peut l'utiliser avec du code JMF existant. Cependant, certaines parties du projet sont encore au cours de développement, donc il peut y arriver qu'on se trouve devant plus de travail à faire.

Pour plus de détails, voir http://fmj-sf.net/

Cette solution produisait des résultats partiellement satisfaisants dont on a pu détecter et distinguer les différentes webcams installées sur la machine, mais le problème qu'on a rencontré est que même si la webcam n'est pas branchée, on reçoive les informations la concernant (Pilotes, formats supportés, …). Nous avons donc laissé tomber cette API, et nous avons découvert l'API DirectShow.

C. DirectShow :

DirectShow (parfois abrégé en DS ou dshow), est une API multimédia développée par Microsoft afin de permettre d'effectuer différentes opérations avec des données médias. Il remplace l'antérieure technologie Vidéo For Windows de Microsoft.

Basé sur le framework Microsoft Windows Component Object Model (COM), DirectShow fournit une interface commune pour les médias dans de nombreux langages de programmation.

- **Architecture de l'API DirectShow :**

DirectShow divise une tâche complexe de multimédia (par exemple : lecture vidéo) en une séquence de traitements fondamentaux connus sous le nom de filtres (la figure(1)). Chaque filtre représente une étape dans le traitement des données, il a des entrées et / ou des broches, des sorties (output pins) qui peuvent être utilisées pour connecter un filtre a d'autres filtres. Le caractère générique de ce mécanisme de connexion permet aux filtres de se connecter de différentes manières afin d'implémenter les différentes fonctions complexes. Pour implémenter une tâche complexe, le développeur doit d'abord construire un filtre graphique en créant des instances de ces filtres, et puis connecter l'ensemble des filtres.

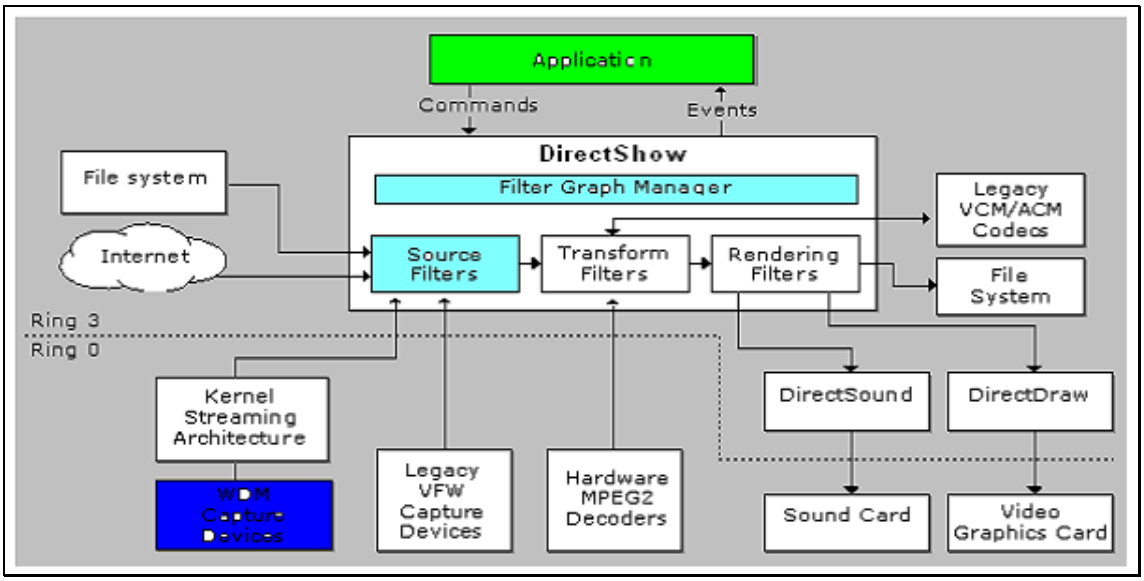


Figure 15.*Architecture de l'API DirectShow*

Comme le montre ce schéma de la figure 15, les filtres de DirectShow communiquent avec une grande variété de dispositifs, y compris le système de fichier local, TV tuner et les cartes de capture vidéo, les codecs VfW, l'affichage de vidéo (par le biais de DirectDraw ou GDI), et la carte de son (par le biais de DirectSound). Ainsi, DirectShow isole l'application de beaucoup de la complexité de ces dispositifs. Elle fournit également des filtres de compression et de décompression de certains formats de fichiers.

Pour plus de détails sur directshow, voir annexe2

- **DSJ :**

DSJ (DirectShow Java wrapper) est une API Java dédiée aux applications multimédias. Elle permet d'utiliser ‘‘en langage Java’’ toutes les fonctionnalités de l'API DirectShow. A titre d'exemple, DSJ permet de lire et d'encoder un très grand nombre de formats média. On peut aussi exploiter tout le matériel (périphériques) que les applications natives y accèdent via DirectShow. Ce projet connait une évolution permanente et est utilisé de nos jours dans différentes applications. La classe principale de cette API est DSFilterGraph représentant les filtres pour DirectShow.

3.4.1.3 Acquisition du media avec DSJ

La Classe DSFiltergraph

- **La hiérarchie et la structure**

java.lang.Object
└java.awt.Component
└java.awt.Canvas
└**de.humatic.dsj.DSFiltergraph**

Elle possède des classes filles dédiées aux différents cas d'utilisation d'un flux multimédia :

DSCapture (que nous avons utilisée) : Pour accéder aux périphériques de capture Audio et Vidéo du système.

DSMovie : Pour le traitement de la vidéo, son édition et son encodage.

DSStreamBufferGraph : Pour les applications interagissant avec la télévision.

DSVcam : Traitement des codeurs standards pour caméras (DVcamcoders).

DsDVD : Lire et interagir avec des Dvds.

DsGraph : Classe de base pour des traitements de bas niveaux.

DSHDVTape : Pour les caméras mpeg HDV et les lecteurs D-Vhs.

Et d'autres …

- **Fonctionnement**

DSFiltergraph fournit différents modes de rendu qui déterminent comment la sortie (output) video du filtergraph sera intégrée dans l'interface graphique (GUI).

La version dsj 0-8-5 fournit des fonctionnalités plus avancées, notamment DirectShowFilters permettent d'insérer '' à partir de java '' des données dans DirectShow, on peut par exemple dessiner au dessus d'une vidéo, la créer, … Ainsi, cette version a introduit un nouveau package dédié à l'implémentation de protocoles pour les applications réseaux, tels que RTSP (Real Time Streaming Protocol) et RTMP (Real Time Messaging Protocol)…

Enfin, DSJ est facile à déployer, ça consiste en deux fichiers : dsj.dll (Dynamic Link Library) et dsj.jar (Java Archive) qui font à peu près 1,2 MB au total. Elle peut être utilisée à tout niveau : local, application web, applet, côté serveur et marche sur toutes les versions récentes de Windows (vista incluse).

NB : la version minimale requise du JRE pour utiliser DSJ est la 1.4.

Pour plus de détails sur l'API DSJ, voir Annexe3.

DSCapture

DSCapture utilise la classe DSFilterInfo pour gérer les dispositifs de capture, leur sorties et les formats qu'ils supportent. La méthode QueryDevices() de la classe DScapture retourne un tableau de 2 lignes :

```
DSFilterInfo[][] dsfi = DSCapture.queryDevices();
```

La première contenant, sous forme de structures, toutes les informations nécessaires sur les périphériques de capture vidéo installés mais aussi branchés sur la machine.

Ces informations comprennent :

PIN ID :

Identifiant associé à chaque périphérique de capture vidéo (webcam), il permet de distinguer les webcams les unes des autres.

PIN name :

nom du pilote du périphérique.

Formats :

tous les formats supportés.

La deuxième contenant les périphériques de capture audio (tel que micro, …) que nous n'utiliseront pas dans notre cas.

Le fragment de code suivant montre comment on crée les objets DSFiltergraph qui vont nous permettre par la suite d'acquérir les flux transitant sur les différentes webcams :

```
public void createDataSources() {

 // récupération des périphériques disponibles
 dsfi = DSCapture.queryDevices();

 nbre_webcams = dsfi[0].length-1;

 //on crée autant d'objets DSFiltergraph que de webcams
 dsfg = new DSFiltergraph [nbre_webcams];

 /** on impose le format 320 * 240 */
 for(int i=0; i< nbre_webcams; i++){

       dsfi[0][i].getDownstreamPins()[0].setPreferredFormat("320 * 240");
 }
 dsfg[i] = new DSCapture(0, dsfi[0][i], false, null, this);
}

/** DSCapture(int flags, DSFilterInfo videoDeviceInfo,
 * boolean captureAudioFromVideoDevice, DSFilterInfo audioDeviceInfo,
 * java.beans.PropertyChangeListener pcl);
 * Creates a DSCapture object that uses the supplied video & audio devices
```

La méthode getImage() définit dans la classe DsFilterGraph retourne l'image instantanée du flux sur la webcam.

```
BufferImage bi = dsfg[numWebcam].getImage();
```

En d'autres termes, cette méthode permet de convertir un flux vidéo à un instant donné en un objet Image.

Ces images seront envoyées sur le réseau et affichées sur l'interface du client (applet), l'une après l'autre et de façon continue, ce qui donnera l'impression de visualiser une vraie vidéo.

3.4.2 Réalisation du module diffusion

3.4.2.1 Objectif

Il est important de savoir dans un premier temps quels sont les besoins de bande passante.
La compression du trafic résulte d'un arbitrage entre le niveau de bande passante, l'espace de stockage nécessaire, la qualité des flux vidéo et les coûts de compression.
Une forte compression allège les besoins en bande passante et en espace de stockage, mais pèse sur la qualité d'image.
Le format MPEG-4 est adapté à la majorité des applications de Surveillance IP. Le MJPEG est néanmoins plus pertinent lorsque l'accent est mis sur la qualité des images (constitution de preuves par exemple) et offre l'avantage d'économiser en matière de caméra vidéo.
La consommation classique de bande passante pour un flux vidéo sera de 8 Ko par trame (MPEG-4 en VGA) et jusqu'à 450 Ko par trame pour du MJPEG en résolution 2084x1536. Les vidéos seront proposées avec 10 à 15 trames par seconde (au minimum) ce qui implique une consommation de 64 Kbps à 34 Mbps.
Une vidéo à très haute résolution risque d'accaparer la totalité de la bande passante.

En résumé : La compression MPEG-4 offre une excellente qualité d'image mais il faut préférer un format plus léger pour un stockage plus important.

A savoir : Si la vidéo surveillance puise dans les ressources du réseau de l'entreprise (elle peut en fonction de la formule adoptée être externalisée), l'engorgement du serveur de stockage risque de paralyser l'utilisation des outils informatiques.

Cette congestion peut survenir suite à un mauvais paramétrage du niveau de compression et du nombre d'images par seconde des caméras. Pour éviter de générer des flux trop importants, il est nécessaire de veiller à la définition de l'image souhaitée en fonction des capacités du réseau emprunté.

3.4.2.2 Fonctionnement

Le fonctionnement est relativement simple, le programme à intervalles réguliers, capture via la où les webcams une image, puis la compresse en MJPEG et enfin l'envoie sur le réseau.

Il suffit ensuite d'afficher une page html à partir du serveur de vidéosurveillance (donc depuis n' importe où dans le monde) qui va afficher l'image correspondante.

1) On choisit un format d'image de taille minimale mais offrant une qualité acceptable:

- VGA: 640*480, format par défaut pour certaines webcams, la taille de l'image est de 16KB envions
- CIF: 320*240, (celui qu'on a choisit), la taille est entre 2 et 3KB.
- QCIF: 160*120, prend 4 fois moins taille que le CIF.

De ce fait on gagnera d'une part en qualité d'image et d'autre part en taille des objets envoyés sur le réseau.

2) Frame Rate (nombre d'images par seconde) : de 25 à 30 images/s.

Remarque : Le taux d'envoi des images affectera d'un côté le serveur en nombre d'instructions exécutées et déterminera, de l'autre côté (client), la qualité de la vidéo. Il va falloir donc faire en sorte de minimiser ce taux tout en gardant une qualité de vidéo acceptable (24 images/seconde au minimum).

3) Enfin la compression : on a choisit le MJPEG.

3.4.2.3 Choix du protocole utilisé pour le transfert du flux vidéo

Un des intérêts majeurs de l'Internet Multimédia est de pouvoir effectuer un Streaming du flux de données.

Le Streaming consiste à découper les données en paquets dont la taille est adaptée à la bande passante disponible entre le client et le serveur. Quand le client a reçu suffisamment de paquets (bufferring), l'application cliente commence à jouer un paquet, décompresse un autre et reçoit un troisième. Ainsi l'utilisateur peut avoir le flux multimédia sans avoir à télécharger tout le fichier. Toutefois, il y a un retard du à la bufferisation.

Version 1 : Utilisation du protocole TCP :

Rappel : La sérialisation consiste à écrire des données présentes en mémoire vers un flux de données binaires.

Un objet Image n'est pas seriablisable, mais un objet ImageIcon l'est. On va donc convertir les images obtenues en ImageIcons pour pourvoir les sérialiser dans le flux a destination de l'Applet. Ensuite pour l'envoi au client via la socket, on utilise la méthode writeObject(icon); de l'ObjectOutpoutStream dans lequel on encapsule les données à envoyer sur la socket. Enfin, du coté applet (client), on récupère les imageIcons, on les convertit en Images et on les affiche sur l'applet l'une après l'autre.

Problème : affichage très lent, non temps réel !

Pour plus de détailles voir l'annexe 4

Version 2 : optimisation de la diffusion avec le protocole RTSP :

- Pourquoi RTSP ?

RTSP (Real Time Streaming Protocol) permet de contrôler la distribution de flux multimédias (streaming) sur un réseau IP. C'est un protocole de niveau applicatif prévu pour fonctionner sur des protocoles tels que RTP/RTCP et RSVP. Les flux peuvent provenir soit de clips stockés soit d'une source temps réel (caméra, micro).

RTSP a été développé par Real Networks, Netscape et l'Université de Columbia. Il est implanté sur les produits de ces sociétés.

- Quelles sont les fonctions de RTSP ?

RTSP offre des fonctions de type magnétoscope à distance (lecture, pause, avance rapide, rembobinage rapide, arrêt...). Il peut être utilisé pour rechercher un média sur un serveur de médias, inviter un serveur de médias à rejoindre une conférence (dans le e-learning par ex), ou ajouter un média à une présentation existante.

RTSP peut être utilisé aussi bien dans des applications unicast que multicast, et peut contrôler et synchroniser plusieurs flux audio ou vidéo. Il ne fournit pas lui-même le flux qui est à la charge d'autres protocoles comme *RTP*. Il n'y a pas de notion de connexion dans RTSP, bien que le serveur maintienne une session ayant un identificateur.

Une session RTSP ne correspond pas à une connexion de transport comme TCP. Durant une session, RTSP peut ouvrir et fermer plusieurs connexions de transport à chaque requête. RTSP est donc basé sur le protocole RTP.

- RTP (Real-Time Transport Protocol)

RTP est un protocole de communication informatique. Ce n'est pas un réel protocole de transfert, puisqu'il utilise l'UDP. Le TCP n'étant pas multicast et ne permettant pas un envoi immédiat de flots de données. Il n'est pas non plus vraiment temps-réel par lui-même (les réseaux actuels comme l'Ethernet n'étant pas temps-réel puisqu'il n'y a pas de délai maximum garanti), mais sera utilisé avantageusement sur un réseau temps réel (par exemple un réseau ATM à bande passante garantie, un canal optique, une radiodiffusion, ou un canal satellite).

Il accorde des fonctions temporelles en tant que service pour des applications multimédia, comme la voix sur IP pour la téléphonie sur Internet ou la diffusion de contenus vidéo en direct. Il ajoute un en-tête spécifique aux paquets UDP pour informer sur le type de média transporté, le séquencement et la synchronisation des datagrammes, afin que le récepteur puisse détecter les datagrammes perdus sur le réseau ou incorrectement reçus, et puisse éventuellement reconstituer un flux continu.

RTP est unidirectionnel et peut être utilisé pour la diffusion (*multicast*). Il est alors extrêmement économique en termes de ressources réseau pour servir un grand nombre de récepteurs, ce qui permet d'augmenter considérablement le débit utile et la qualité de codage du contenu.

Il peut éventuellement être utilisé conjointement avec un canal de retour (*feedback*) sur la qualité de service (QoS) via RTCP (*Real-Time Transport Control Protocol*), négocié indépendamment (voir RTSP). Ce *feedback* peut par exemple informer l'émetteur sur les propriétés temps-réel du canal, l'état du tampon du récepteur, ainsi que demander des changements de compression/débit pour les applications multimédia par exemple (dans ce cas, les données manquantes pourront être transmises via *Unicast*).

Pour la diffusion en masse cependant (flux en direct, radiodiffusé), cette voie de retour n'est généralement pas utilisée, mais le contenu est transmis plusieurs fois en parallèle avec un

décalage temporel suffisant pour pallier les interruptions temporaires de qualité de réception, mais n'excèdent pas les limites des tampons des récepteurs (normalement pas plus d'une quinzaine de secondes d'écart). Le récepteur peut alors reconstituer et réordonner la séquence complète afin d'obtenir un flux continu sans perte.

Pour les contenus protégés à valeur ajoutée, l'absence de voie de retour implique l'utilisation de clé de déchiffrement du contenu, que le récepteur doit négocier séparément avec l'émetteur (chacun peut recevoir facilement le contenu chiffré simplement en se connectant au routeur de diffusion). RTP lui-même ne s'occupe pas du chiffrement et transporte le contenu de façon transparente.

RTP est la version normalisée internationale de l'ancien protocole propriétaire RDP (initialement créé pour *Real Player*) en voie d'obsolescence.

Le protocole SRTP (acronyme de Secure Real-time Transport Protocol) est le pendant sécurisé (chiffré) de RTP.

3.4.3 Réalisation du module intrusion

3.4.3.1 Introduction au traitement d'images

3.4.3.1.1 Présentation

Depuis la fin des années 90, la numérisation des contenus et la progression de puissance des ordinateurs ont rendu possible le traitement en temps réel des images de la vidéo pour en extraire des interprétations (que voit-on à l'image, que se passe-t-il, qui va où, etc.). D'abord effectués en noir et blanc, puis en couleur, ces traitements ont commencé à sortir des laboratoires de recherche dans cette période, et ont constitué des solutions exploitables, d'abord pour la surveillance routière, puis pour la surveillance de personnes et d'objets, et aussi pour la biométrie faciale, etc.

Une séquence vidéo numérique peut être lue et manipulée par un programme sur un ordinateur en tant que flot d'images annotées (date, numéro d'image, …).

Chaque image de ce flot est constituée de pixels (terme issu de la contraction des mots anglais « picture elements ») qui constituent autant de points caractérisant la taille/résolution de l'image. Les capteurs d'images à l'origine de la vidéo caractérisent le nombre de pixels de largeur et de hauteur des images qu'ils engendrent; ce sont d'ailleurs des paramètres importants lors du choix d'un capteur puisqu'ils vont conditionner la résolution à laquelle on voit les détails de l'image, une fois numérisée.

On trouve sur le marché des capteurs produisant des images vidéo de tailles très variées par exemple 160x120, 320x200, 288x352 (nommé CIF), jusqu'à 800x600, 576x704 (4CIF), 1000x100. Plus il y a de pixels, plus riche est l'information produite. Chaque pixel est en général représenté par une ou quelques valeurs entières qui codent son intensité (en noir et blanc et en caméra thermique/infrarouge) et, dans le cas de la couleur, sa chromaticité et sa saturation.

Les représentations les plus couramment utilisées dans les traitements numériques de l'image sont le mode RGB (pour lequel un pixel est représenté par trois entiers dont la valeur caractérise le poids des couleurs Rouge, Vert (Green) et Bleu associées au pixel) et le YUV (pour lequel il s'agit de la chromaticité, la saturation et l'intensité).

3.4.3.1.2 Comment comparer deux images

Pour comparer deux images d'une vidéo, on s'appuie simplement sur le fait que les images successives d'une vidéo sont la plupart du temps toutes de la même taille, et donc on compare un à un les pixels dans l'ordre « ligne x colonnes ».

Tous les pixels comportant une valeur différente d'une image à l'autre appartiennent à la différence entre les deux images. Cependant, comme les capteurs sont des équipements physiques imparfaits, les images successives d'un même plan vidéo (même éclairage, caméra immobile, …) présentent en général de l'une à l'autre de très petites différences dues à des incertitudes ou des erreurs de mesure effectuées par le capteur : tous ces aléas ont été regroupés, quelle que soit leur cause, dans un terme qu'on appelle « bruit ».

Le bruit d'un capteur dépend de ce capteur mais aussi de l'optique, des composants électroniques de la camera, de la chaîne de numérisation utilisée, jusqu'à disposer de l'image numérique, qui une fois élaborée est transportée avec des protocoles assurant son intégrité par des mécanismes de vérification (checksum), et de répétition en cas d'erreur à la transmission.

Pour en revenir à la différence entre deux images, en général les écarts constatés par une constante (appelée seuil) placée très au dessus de la valeur moyenne du bruit, ce qui permet de détecter des différences plus certaines entre les deux images, c'est-à-dire des différences qui sont valides dans le monde réel, au-delà du bruit du capteur.

3.4.3.1.3 La détection de mouvement, les différentes techniques:

Les caméras numériques présentes sur le marché coûtent de moins en moins cher, et leur encombrement est de plus en plus réduit. Parallèlement, la puissance de calcul des ordinateurs actuels permet d'envisager sérieusement le traitement automatique en temps réel de séquences vidéo. C'est pourquoi les industriels ont aujourd'hui tendance à opter pour des solutions à base de vision artificielle pour résoudre des problèmes qui étaient auparavant traités par d'autres procédés, tels que la surveillance par un opérateur humain ou l'utilisation de capteurs plus mécaniques. Quelle que soit l'application, la première tâche d'un système d'analyse de séquences vidéo est toujours la détection de mouvement, et si possible, la détection des objets (segmentation) mobiles.

La difficulté de cette tâche est très variable selon les conditions d'acquisition, la précision et la rapidité du traitement escomptées. Une liste relativement exhaustive des difficultés liées à l'acquisition et au contenu de la scène.

Pour détecter un mouvement dans une séquence vidéo, on peut utiliser différentes techniques qui sont toutes issues de la recherche dans les disciplines du traitement d'images, du traitement du signal, et de l'intelligence artificielle :

- ***La technique du flot optique*** (Horn et Schunck, 1981) : permet la mise en œuvre d'une analyse globale du mouvement à l'aide d'une équation reliant la variation d'intensité lumineuse en un point avec la vitesse de déplacement de ce point. Cette technique permet d'analyser des scènes dont la totalité de l'image est en mouvement, et d'y distinguer des objets en mouvement relatif les uns par rapport aux autres. Cette technique est utilisée en météo (analyse et mesure des mouvements des nuages, des cyclones), et en aide à la conduite de véhicules (détection et analyse des objets mobiles devant le véhicule : autres voitures, piétons…).

- ***La technique de l'«image de fond»*** (Wren, Pentland, 1996) : se limite aux caméras en position fixe et permet, grâce à la mise en place et à la mise à jour permanente d'une image du fond vide, de distinguer des objets mobiles par différence à ce fond.

- ***La technique des points caractéristiques*** (C. Schmid, R. Mohr, 1997) : se concentre sur la recherche de points caractéristiques dans l'image (J. Harris, 1988, points de l'image aux caractéristiques fortement marquées : coins, bords, puis par extension centres de régions) et recherche leur correspondants d'une image à l'autre pour en déduire un mouvement, et pour regrouper ensemble les points proches ayant un mouvement cohérent.

Toutes ces techniques sont sensibles au bruit du capteur et nécessitent des traitements complémentaires, comme par exemple le filtrage, pour fonctionner de façon satisfaisante.

3.4.3.2 Identification de l'intrusion : Implémenter la technique d'image de fond (background subtraction).

Le modèle le plus simple pour identifier une intrusion consiste à considérer à chaque instant t que l'image au temps t-1 représente l'image de fond, et que les zones en mouvement sont celles qui ont changé d'apparence entre t-1 et t. En d'autres mots, il s'agit de comparer un pixel i de l'image

au temps t avec le pixel i de l'image de fond au temps t-1, s'il y'a une différence (supérieure au seuil) entre ces deux pixels ca veut-dire qu'il y'a un changement.

Afin d'analyser et comparer les images, on va se baser sur les niveaux de rouge, vert et bleu (RGB) de chaque pixel de l'image.

Pour récupérer les pixels d'une image on utilise un objet non spécifique à DSJ mais qui est fournit par le JDK, le PixelGrabber, voir Annexe5

A partir de cet objet, on va remplir un tableau de pixels ayant la taille de l'image (width * height).

Le fragment de code ci-dessous montre clairement comment ça marche.

```
/* Create a PixelGrabber object to grab the (x, y, w, h) rectangular
* section of pixels from the specified image into the given array. */
grab1 = new PixelGrabber(img1, 0, 0, 320, 240, pixels1, 0, 320);
grab2 = new PixelGrabber(img2, 0, 0, 320, 240, pixels2, 0, 320);
/* Request the Image or ImageProducer to start delivering pixels and
wait for all of the pixels in the rectangle of interest to be
delivered.*/
try {
// we grabbe to be sure that the Image has finished loading
grab1.grabPixels(); //: boolean
grab2.grabPixels();
}
catch (InterruptedException ie) {
System.err.println("impossible de grabber les pixels de l'image");
}
// on recupere la moyenne des 3 couleurs pour les 2 images
getColors(pixels1, colors1);
getColors(pixels2, colors2);
//Actions a executer en cas de mouvement
if (imageChanged(colors1, colors2)){
//Changement détecté entre img1 et img2
```

La méthode getColors calcule les moyennes des trois couleurs R, G et B et les stocke dans un tableau colors.

```
/* on cumule les niveaux de chaque couleur en parcourant le tableau pixels */

Color     pixelColor = new Color(pixels[i]);
Int red = red + pixelColor.getRed();

//on calcule la moyenne de chaque couleur
red = red / imageSize;
colors[0] = red
```

La méthode ImageChanged vérifie si la différence entre les couleurs de l'image1 et celles de l'image2 est supérieure au seuil, et retourne « true » le cas échéant.

3.4.3.3 Gestion des intrusions

Concernant la gestion des intrusions de notre application, nous avons choisi de développer plusieurs solutions. Nous verrons dans un premier temps l'enregistrement de séquences vidéo. Dans un second temps, nous vous exposerons comment se déroule l'envoi de SMS. Enfin, nous nous pencherons sur l'envoi d'email.

Enregistrement de séquences vidéo :

Une première nécessité et de pouvoir garder une trace de tout ce qui s'est passé lors de l'intrusion. En d'autres termes, avoir la possibilité de sauvegarder des séquences vidéo qui pourront servir de preuves au cas de vol par exemple.

A l'instar de l'API JMF, DSJ nous permet d'enregistrer de la vidéo depuis une webcam. Cependant, on ne peut pas s'en servir (de DSJ), car la webcam sera déjà en cours d'utilisation par l'application sur le serveur de vidéosurveillance et donc ne peut pas être utilisée par deux applications simultanément.

Solution : exploiter les images capturées et envoyées sur le réseau.

Dès la détection d'un mouvement, on continue (*) la sauvegarde des images dans un répertoire pendant une durée donnée, ensuite on génère une vidéo depuis ces images.

Un autre problème était le fait de rater les (quelques) premières images qui présentent un changement, et donc la vidéo enregistrée ne commencera pas exactement par la 1ère image montrant le 1èr changement.

Solution : L'objet ImagesComparator chargé de détecter les mouvements fonctionne comme suivant :

Une fois activé, il enregistre un certain nombre d'images (une vingtaine par exemple) successives et n'en compare que les 6 dernières.

Deux cas peuvent se présenter :

- S'il n'y a pas d'intrusion, il écrase les images par d'autres nouvelles et recommence le processus.
- Sinon, il annonce l'intrusion et continue (*) l'enregistrement d'images. De ce fait, les 20 images prises auparavant vont servir à avoir à peu prêt 200 ms avant l'intrusion et donc on verra ce qui s'est passé depuis le début.

Envoi d'SMS :

Pour l'envoi d'SMS depuis une application java, une solution est d'avoir un abonnement avec un operateur téléphonique fournissant le service d'envoi d'SMS via internet.

- ***Comment ça marche ?***

On leur envoi un e-mail contenant les données de l'SMS (texte et destinataires) et puis leur service s'occupe de transmettre le message à ses destinataires.

Envoi d'e-mail en utilisant l'API JavaMail :

JavaMail est une technologie java pour envoyer et recevoir des emails.

Ce n'est pas un serveur de mails, mais un outil pour interagir avec le serveur de mails. Les applications développées avec JavaMail peuvent être ainsi comparables aux différentes messageries que l'on rencontre tel qu'Outlook, Lotus, Eudora...

C'est une API qui permet donc d'utiliser le courrier électronique (e-mail) dans une application écrite en java (application cliente, applet, servlet, EJB...).

JavaMail est très facile à utiliser, elle fournie une souplesse qui permet de la faire évoluer et de rester le plus indépendant possible des protocoles utilisés

Pour envoyer ou recevoir des messages, JavaMail utilise différents protocoles comme smtp, Imap, Mime, MNTP... Dans notre cas on s'intéresse seulement au protocole SMTP qui permet de transférer le courrier vers un serveur de messagerie électronique.

La seule exigence est donc de disposer d'une adresse de serveur SMTP.

Deux possibilités :

- Demander l'adresse du serveur SMTP au FAI (fournisseur d'accès à internet).
- Installer un serveur SMTP sur la machine (exemple : QuickServerSMTP).

3.4.4 Réalisation du module authentification et redirection

La communication et la diffusion de la vidéo sur réseau local ou public sont sécurisées et la connexion ou caméra réseau s'établie en trois étapes :

Authentification

L'utilisateur doit s'identifier sur le réseau local (intranet) ou à distance (internet). Pour ce faire, certaines données d'identité sont communiquées au réseau ou au système, comme par exemple un nom d'utilisateur et un mot de passe.

Autorisation

Autoriser et accepter l'authentification automatiquement, puis vérifier l'identité de l'utilisateur par rapport aux informations contenues dans la base de données de la caméra réseau. Si l'utilisateur est authentifier alors il aura l'accès pour visualiser la vidéo en temps réel sur sont PC distant.

Confidentialité

VPN et SSL sont les méthodes utilisées pour assurer la confidentialité et la sécurisation du canal qui transporte les données vidéo. Tableau comparatif entre la vidéosurveillance traditionnel et la vidéosurveillance sur IP : Une infrastructure reposant sur la technologie IP offre aux utilisateurs une multitude d'avantages réduisant ainsi les coûts d'investissement liés à la vidéosurveillance.

3.5 PRESENTATION DU PROTOTYPE

3.5.1 Accès au système : l'interface web

Voici un aperçu de l'interface web du système Rapace. Le site se compose de 5 onglets principaux :

- ➔ Accueil : page d'accueil du site
- ➔ Notre service : présentation des fonctionnalités du système
- ➔ F.A.Q : foire aux questions
- ➔ Mon Compte : à partir de cette page, le client se connecte à son compte et peut accéder à ses webcams.
- ➔ Inscription : permet de s'inscrire au système Rapace

Figure 16. *Page d'accueil du site web*

Dans les parties suivantes, nous allons vous dévoilé des captures d'écrans de notre système.

3.5.2 La connexion avec le serveur de vidéosurveillance

Voici la figure qui montre l'interface principale de notre application en utilisant deux webcams

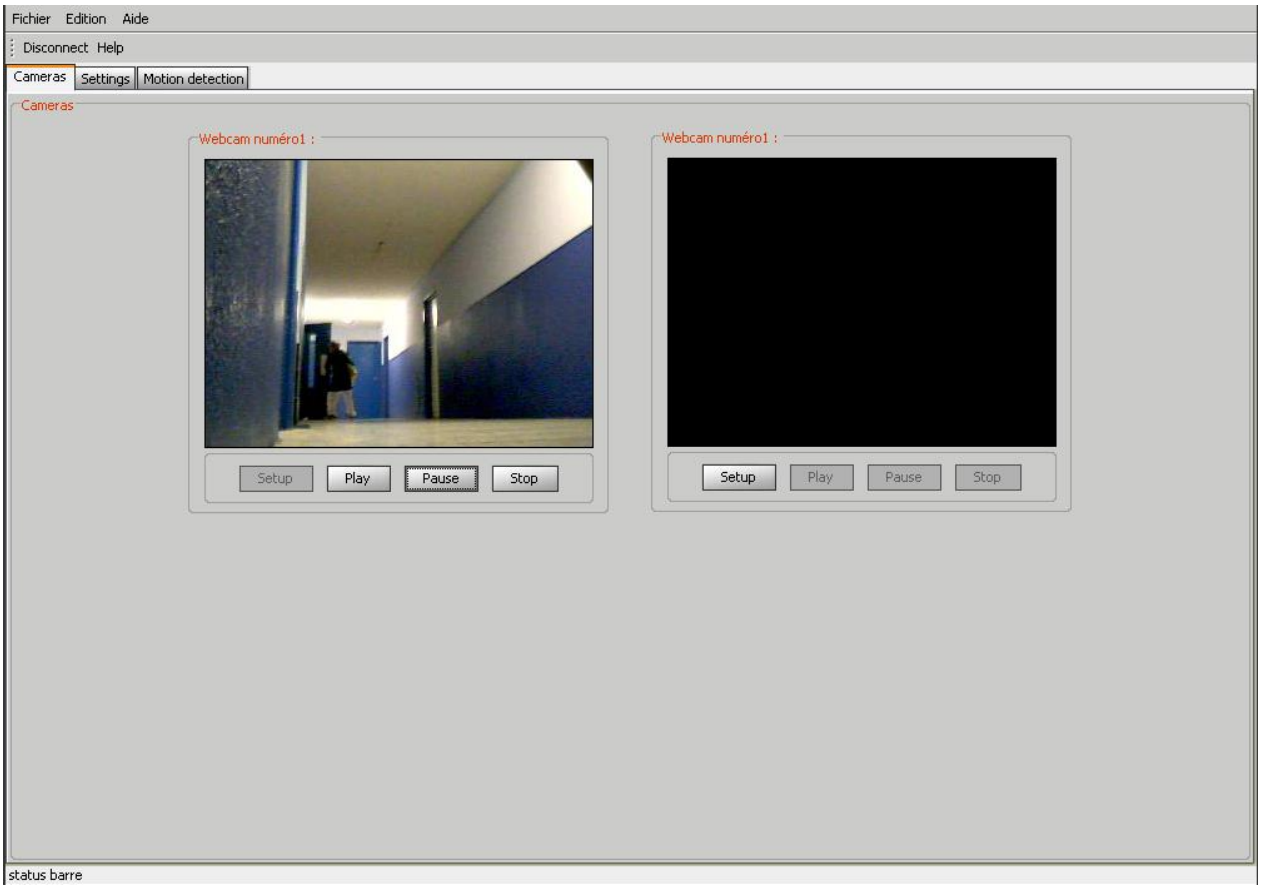


Figure 17.*Interface principale en utilisant 2 webcams*

Figure de l'interface principale de notre application en utilisant 4 webcams.

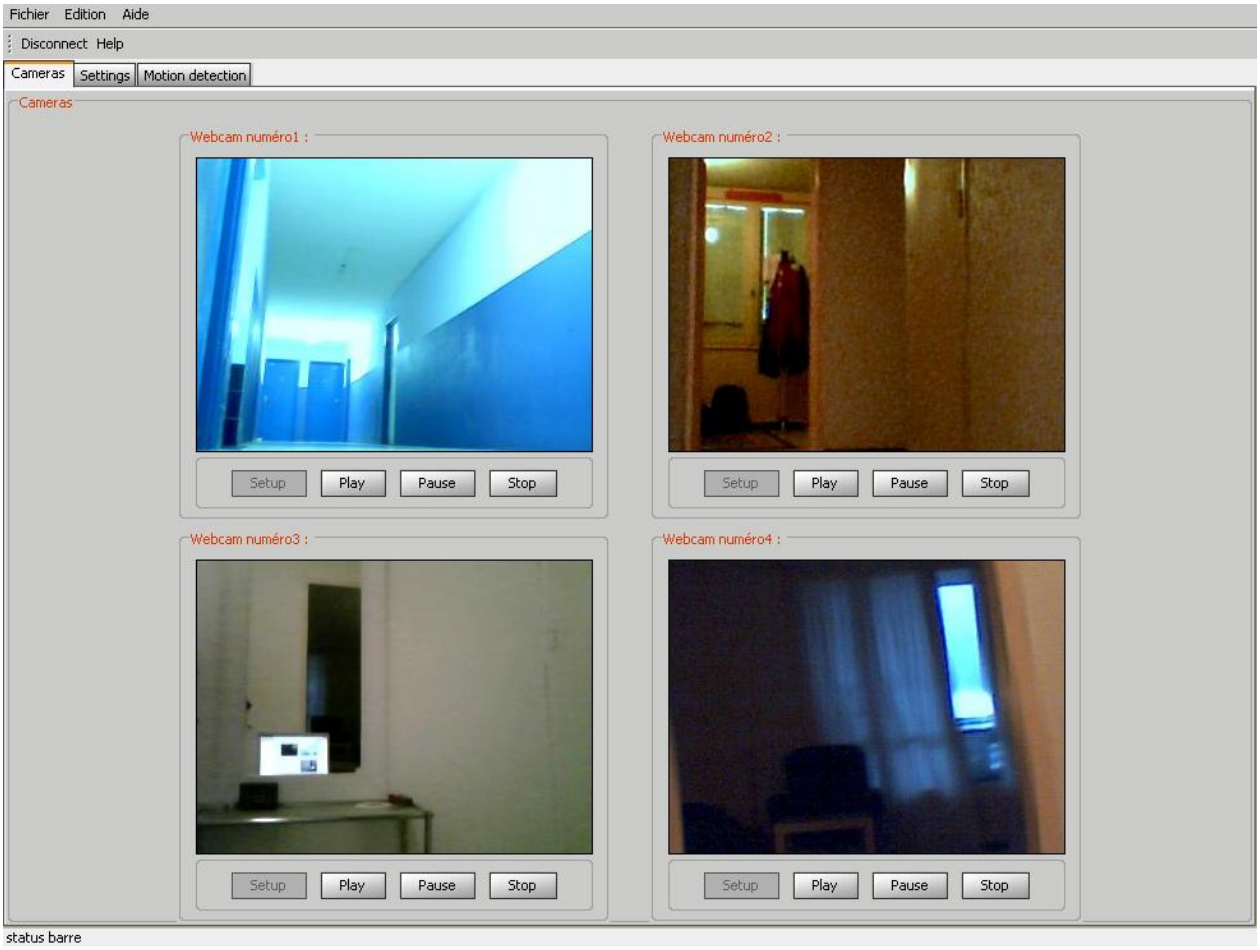


Figure 18.*Interface principale en utilisant 4 webcams*

3.5.3 Le paramétrage

Figure qui montre l'onglet de paramétrage de notre application.

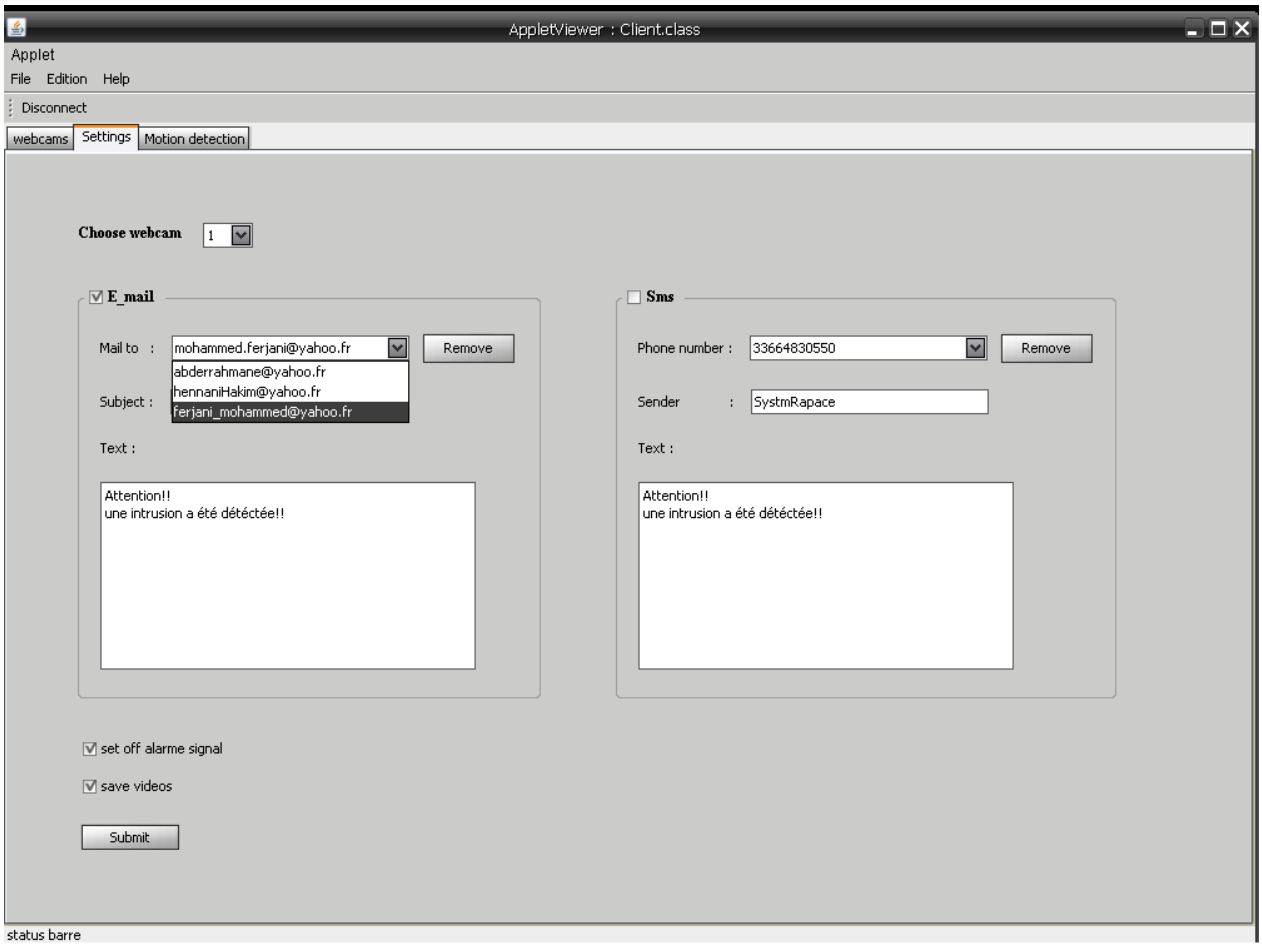


Figure 19.*Onglet de paramétrage*

3.5.4 Interface de gestion des intrusions

Figure qui montre l'onglet à partir duquel on peut activer ou désactiver la détection de mouvement en précisant le niveau de sensibilité.

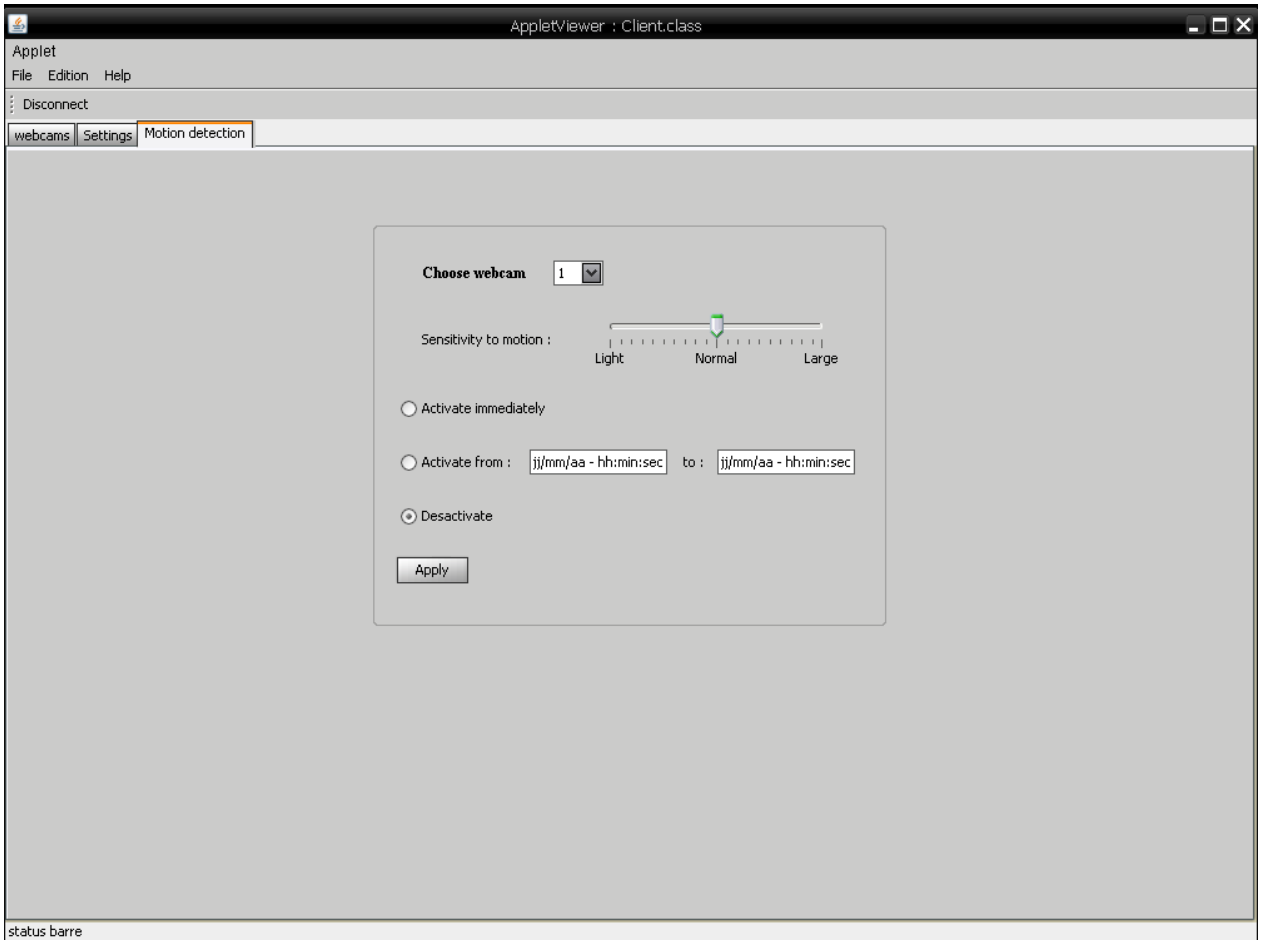


Figure 20. *Onglet de gestion d'intrusion*

3.6 PERSPECTIVE D'AMELIORATION

3.6.1 Webcam Versus camera IP

Le système Rapace fonctionne à l'heure actuelle avec des webcams. Cependant, nous avons la possibilité dans le futur de pouvoir faire évoluer ce système de vidéosurveillance en remplaçant les webcams par des cameras IP.

Il est important de se pencher sur la distinction entre une webcam et une camera IP. Les caméras IP n'ont pas grand chose à voir avec les webcams que l'on trouve depuis plusieurs années. Les webcams permettent aux internautes de pouvoir utiliser la vidéo lors de leurs communications par internet (msn messenger, skype etc.). Toutefois, ces webcams si populaires sont extrêmement différentes des caméras IP (aussi appelées Caméras Internet ou réseau). La différence la plus importante est liée au mode de connexion de la caméra elle-même. En effet, l'utilisation d'une webcam traditionnelle sur internet sous-entend d'être relié à un ordinateur (très souvent par une prise USB)

Pour que la Webcam puisse être utilisée, il faut ensuite nécessairement que l'ordinateur auquel elle est rattachée soit allumé.

A contrario, aucune utilisation de cette webcam ne sera possible si l'ordinateur, pour une raison pour une autre, est éteint.

Les fonctions intégrées dans la webcam sont extrêmement réduites puisqu'elles peuvent se reposer sur l'intelligence présente dans l'ordinateur. Ainsi, lorsque vous utilisez msn ou skype, la webcam envoie de l'image vers le PC et c'est ensuite ce dernier qui fait tout le reste, du réglage à l'affichage de ces images.

Tout au contraire, les caméras IP offrent des fonctions vidéo combinées à une intelligence proche d'un ordinateur. Par conséquent, les caméras réseau peuvent fonctionner sans aucun ordinateur et offrir des fonctions de vidéosurveillance impossibles pour une webcam. Il leur faudra juste une connexion à un réseau local ou étendu (tel qu'internet).

Enfin, et c'est peut être une des raisons de la confusion entre une webcam et une caméra réseau, c'est que cette dernière intègre un serveur Web. Cela signifie que l'on peut accéder à une caméra réseau à partir d'un navigateur internet.

Malgré l'intérêt énorme que portent les médias aux caméras numériques ou de réseau, la majorité des caméras vendues sont toujours analogiques et ce principalement en raison du prix. Souvent a lieu une transition progressive vers l'IP au moyen de solutions hybrides selon lesquelles les caméras analogiques sont raccordées à un réseau IP au moyen d'encodeurs vidéo. A l'heure actuelle, cette combinaison représente toujours une économie de quelque 15% pour les caméras les plus avancées. La tendance est toutefois en train de s'inverser rapidement au profit des caméras IP et la conversion totale est prévue pour les environs de 2010.

Pour être clair, une caméra IP ou de réseau est bien plus qu'une webcam. Une caméra IP comporte son propre microprocesseur et assure elle-même, par analogie avec un DVR, les fonctions de digitalisation, de compression et éventuellement l'analyse des images vidéo.

Les cameras IP présentent de nombreux avantages par rapport à la webcam :

- Elles peuvent être installées partout où un réseau informatique est disponible (câblage ou sans fil). Une fois que ces caméras sont enregistrées dans le réseau, on peut les brancher facilement ailleurs sur le réseau. La technologie analogique ne permet pas une telle flexibilité.

- Elles utilisent l'équipement PC standard, ce qui permet d'ajouter facilement de la puissance de traitement ou de la capacité de stockage supplémentaire, en fonction de la nécessité.

- L'accès aux images peut parfaitement se faire à distance sur le réseau, ce qui permet de confier la surveillance à des opérateurs professionnels plus qualifiés dans une salle d'écrans vidéo et avec un nombre moins important.

3.6.2 Autres

Il est important de souligner le fait qu'il y ait d'autres perspectives d'évolution de notre système.

Nous pouvons ajouter lors de futures évolutions du système Rapace une section vidéoconférence. Cet ajout enrichirait le système de part les avantages de la vidéoconférence. Tout d´abord, elle permettait un gain de temps. En effet, lors des réunions classiques, les participants dépensent beaucoup de temps à se déplacer. Cet avantage est loin d´être négligeable, car il permet d´être plus productif dans le sens où la personne n´aura pas à s´attarder dans les transports publics, pour les réservations, et évitera les arrangements horaires. Le temps épargné sera utilisé au bénéfice des différentes rencontres.

Comme autre avantage, il est important de souligner les économies réalisées. En effet, le fait d´utiliser la vidéoconférence évitera les coûts relatifs aux transports, ainsi que les frais annexes, comme par exemple les frais d´hôtel et les frais administratif.

De plus, il n'est pas utopique d'ajouter à notre système un module de vidéosurveillance mobile ; chose que nous n'avons pu réaliser par manque de temps. Pour ce faire, nous pouvons utiliser l'API J2ME : voir l'annexe 7.7.

4 IMPRESSIONS ET ACQUIS DU TER

4.1 POINT DE VUE TECHNIQUE

La réalisation de ce TER nous a permis de nous familiariser et d'approfondir certaines notions techniques.

Au niveau de la programmation du système, nos connaissances en JAVA ont été compléter par l'utilisation d'API. De plus, au niveau du réseau, nous avons découvert de nouveaux protocoles qui nous ont permis de résoudre des problèmes au niveau du système. Enfin, le développement web nous a permis de se perfectionner dans différent langage tels que le HTML/CSS, le PHP et de découvrir, au niveau du fonctionnement, le langage Flash.

Ce TER à été enrichissant d'un point de vue technique car il nous a permis d'utiliser dans la pratique nos connaissances dans différents domaine de l'informatique (programmation, réseau, développement web) et de découvrir de nouvelles notions.

4.2 POINT DE VUE ORGANISATIONNEL

Ce TER nous a permis de travailler dans des conditions nouvelles. En effet, nous nous sommes placés comme si nous travaillons dans une entreprise avec comme chef de projet notre encadrant. Celui-ci nous à très bien repartis les taches à effectuer, ce qui à contribuer à une lisibilité et une clarté dans notre projet. Chaque membre du groupe avait une tache prédéfinie avec des objectifs à réaliser.

De plus, la planification des réunions (à raison de une par semaine) n'a été que bénéfique pour l'avancement du projet. Chaque membre y effectuait un compte rendu et une présentation de son travail. Il s'en suivait des discussions pour résoudre des problèmes rencontrés et avancer sur l'évolution du projet.

5 CONCLUSION

Nous tirons globalement de ce projet un bilan très positif, bien que nous ayons eu à faire face à des difficultés. Notre capacité à les résoudre et les méthodologies que nous avons employées pour les résoudre sont finalement des motifs de satisfaction.

De plus, la réalisation de ce projet nous a énormément aidés à développer notre créativité et notre imagination, et surtout d'acquérir l'esprit du travail en groupe. Elle nous a aussi permis de mieux découvrir la puissance du langage JAVA et la richesse de ses librairies.

Ce projet a été l'occasion de mettre en pratique la formation théorique que nous avons reçue durant notre parcours universitaire, qui s'est révélée adaptée aux compétences souhaitées. De plus, ce travail qui ne nous a pas été sans peine, constitue aussi un honneur pour nous.

Vous avez donc pu constater qu'il est possible de réaliser un système de vidéosurveillance à distance sans forcément investir de grosses sommes d'argents dans du matériel onéreux comme des camera IP. Toutefois, nous avons atteint notre objectif de réalisation d'un système de vidéosurveillance à distance, avec un mode d'utilisation simplifié pour l'utilisateur, cependant quelques améliorations restent possibles. Par exemple, nous aurions aimé si nous avions eu plus de temps d'ajouter au système un module de vidéosurveillance mobile.
Enfin nous souhaitons que le travail présenté ait une utilité quelconque pour les formateurs ou tout autre lecteur qui y trouveront certains renseignements qui pourront servir d'une manière ou d'une autre.

6 BIBLIOGRAPHIE

6.1 OUVRAGES

- Claude Delannoy, « Programmer en JAVA », 3ème édition, Paris, 2001, 685 pages.
- G. Pujolle, « Les Réseaux », Eyrolles 5ème édition, 2004, 1094 pages.
- Cours de services et qualités des réseaux avec monsieur CLEMENT Saad, université Montpellier 2.

6.2 SITES WEB

- *http://java.sun.com/*
- *http://www.humatic.de/htools/dsj.htm*
- *http://www.java2s.com/*
- *www.Javafr.com*
- *www.developpez.com*
- *www.labo-sun.com*
- *http://fr.wikipedia.org/wiki/videosurveillance*

7 ANNEXES

7.1 ANNEXE1 : JMF

Voir les liens :

http://www.labo-sun.com/resource-fr-articles-998-1-java-j2se-jmf-java-media-framework.htm

http://java.sun.com/javase/technologies/desktop/media/jmf/

7.2 ANNEXE2 : DIRECSHOW

7.2.1 PRINCIPES DIRECTEUR DE DIRECTSHOW :

7.2.1.1 Le graphe de filtres

Le graphe de filtres constitue la base de toute programmation sous DirectShow. Il permet de gérer les flux de données de manière simple en structurant les traitements (appelés filtres) sous forme d'un graphe orienté non cyclique:

- · Les noeuds du graphe représentent les traitements à effectuer.
- · Les arêtes (orientées) du graphe représentent la direction des flux entre les filtres.

Il est ainsi possible d'effectuer des graphes de traitement complexe avec une simplicité relative.

7.2.1.2 Rôle du graphe :

- assembler les filtres (les connecter en veillant à la compatibilité des E/S).
- gérer les filtres (en acquièrant sur ceux-ci les interfaces nécessaires à leurs manipulations).
- contrôler et synchroniser les flux dans le graphe.

Le graphe de filtres est donc un médiateur entre le programmeur et les données. Le programmeur opère donc à un niveau assez élevé de développement. Il n'a pas à se soucier de la manipulation bas-niveau de données audio et vidéo.
Le graphe est un objet de type COM.

7.2.1.3 Les filtres

Conceptuellement, un filtre peut être considéré comme une boîte noire:

- qui appartient à une classe spécialisée en fonction du travail à accomplir.
- avec des entrées et/ou des sorties.
    - chacune de ces entrées/sorties est conceptualisée sous forme d'une borne de connexion (pin).
    - une borne d'entrée permet de recevoir des données depuis un autre filtre.
    - une borne de sortie permet d'envoyer des données vers un autre filtre.

- avec une ou plusieurs interfaces spécialisées, associées à la classe du filtre qui permettent de contrôler et de configurer le comportement du filtre.
- la configuration interne du filtre est assurée par une interface graphique (interne) fournie par le filtre lui-même.

On distingue trois principaux types de filtres:

- · Les filtres sources.
- · Les filtres de transformation.
- · Les filtres de rendu.

Les filtres sont des objets COMs: la classe d'un filtre est définie par sa CLSID, les interfaces sont définies par leurs IIDs.

7.2.1.4 Les filtres sources

Un filtre source n'a aucune borne d'entrée, et une ou plusieurs bornes de sortie.
Ce type de filtre produit un flux de données à partir:

- · d'un support numérique (fichier, CD, DVD, …)
- · d'un périphérique d'acquisition (micro, webcam, caméscope numérique, tuner TV, etc…).
    - S'il est directement supporté par Windows, le filtre correspondant existe déjà.
    - Pour tout nouveau matériel, à partir du moment où son pilote (driver) répond au modèle WDM (Windows Driver Model), celui-ci installe également les filtres DirectShow nécessaires à l'utilisation du périphérique.
    - d'une connexion réseau (streaming).

Il faut au minimum un filtre de ce type dans un graphe (pour « nourrir » le graphe avec des données). On peut utiliser autant de filtres d'entrés que de sources nécessaires à l'exécution de la tâche.

7.2.1.5 Les filtres de transformation

Un filtre de transformation modifie le flux de données qui le traverse. Il reçoit ses données en provenance d'un ou de plusieurs autres filtres (sources/transformation). Il transmet le flux transformé vers un ou plusieurs autres filtres (transformation/rendu).
Il a donc au moins une borne d'entrée et au moins une borne de sortie.
Les filtres de transformations les plus couramment utilisés sont les suivants:

- parser = séparer les données contenue dans un flux pour former plusieurs flux de sortie
- (exemple: séparer l'audio de la vidéo).
- · tee = dupliquer le flux d'entrée en plusieurs flux identique de sortie.
- · multiplexage = combiner plusieurs flux en un seul (combiner un flux vidéo et un flux audio
- pour ensuite les enregistrer dans un ficher par exemple)
- · codec = convertir les données d'un flux en un autre format (compression / décompression).
- · sous-titrage = ajouter une couche de sous-titres à un flux vidéo.
- · montage et transition
- · …

L'installation de nouveaux Codecs rend disponible les méthodes de compression et décompression associées sous forme de nouveau filtre. Dans certains cas, il peut être nécessaire d'enregistrer manuellement ces filtres.

7.2.1.6 Les filtres de rendu

Un filtre de rendu est placé en fin de chaîne de traitement et permet de terminer le traitement:

- par un rendu:
    - vidéo : création automatique (ou manuelle) d'une fenêtre (avec son handle) et rendu du flux dans la fenêtre en utilisant DirectDraw.
    - audio : rendu automatique en utilisant DirectAudio.

  Toutes des ressources nécessaires sont automatiquement allouées et désallouées, la synchronisation

vidéo/audio est assurée par le graphe.

- par l'enregistrement dans un fichier.
- par sa transmission sur le réseau.

Ce type de filtre ne possède que des bornes d'entrée.

Un graphe doit contenir au moins un filtre de rendu.

On peut éventuellement multiplier le nombre de filtres de rendu:

- l'audio et la vidéo sont rendus par deux filtres de rendu différents (un filtre de rendu audio
- et un filtre de rendu vidéo).
- · on peut effectuer un rendu vidéo tout en enregistrant ce même flux dans un fichier.
- · deux flux vidéo issus de deux chemins différents dans le graphe peuvent être rendu dans
- deux fenêtres différentes.

7.2.1.7 Construction d'un graphe à partir de filtres

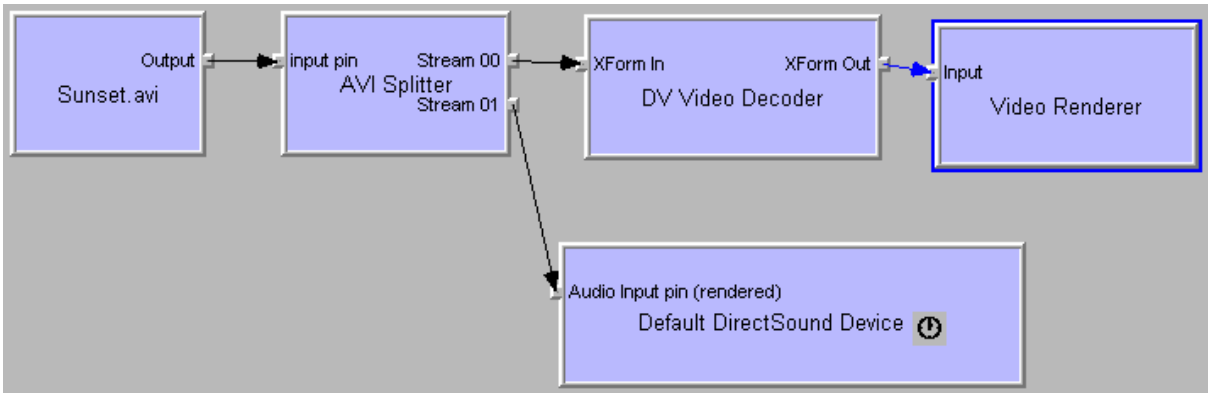
Pour construire un graphe, les filtres peuvent être assemblés avec une grande liberté à partir du moment où les contraintes suivantes sont respectées:

- · toute branche d'un graphe commence par un filtre d'entrée et se termine par un filtre de
- sortie (cohérence du graphe).
- · les comptabilités entre les entrées et les sorties sont respectées ;

Chaque entrée/sortie n'accepte que certains types et format de flux de données. Autrement dit une connexion entre deux filtres n'est possible que si il existe un pin de sortie sur le premier filtre et un pin d'entrée sur le sortie tels que:

- le type du flux de données est le même (audio, vidéo, …)
- le format du flux en sortie est compatible avec le format du flux en entrée.

7.2.1.8 Exemple de graphe des filtres



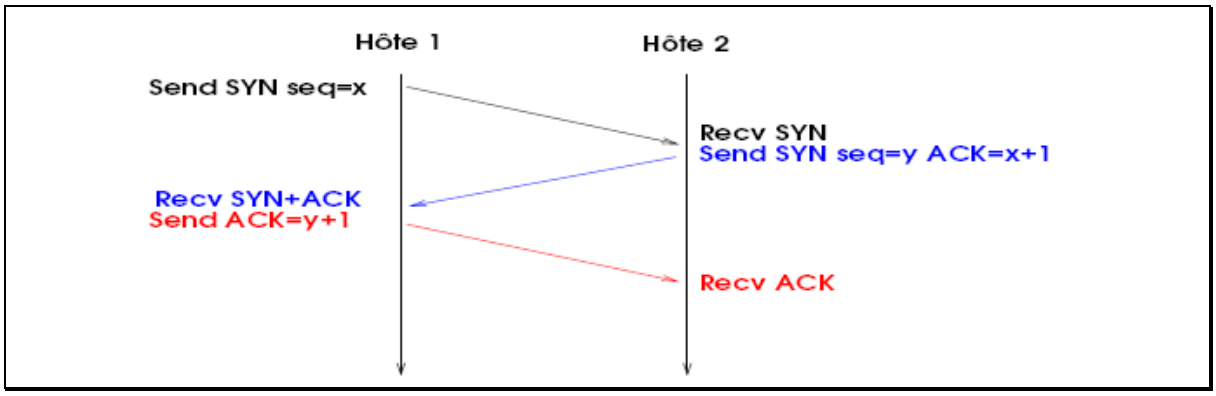
7.3 ANNEXE3 : DSJ

Voir le lien : http://www.humatic.de/htools/dsj.htm

7.4 ANNEXE4 : TCP

7.4.1.1 Etablissement de la connexion :

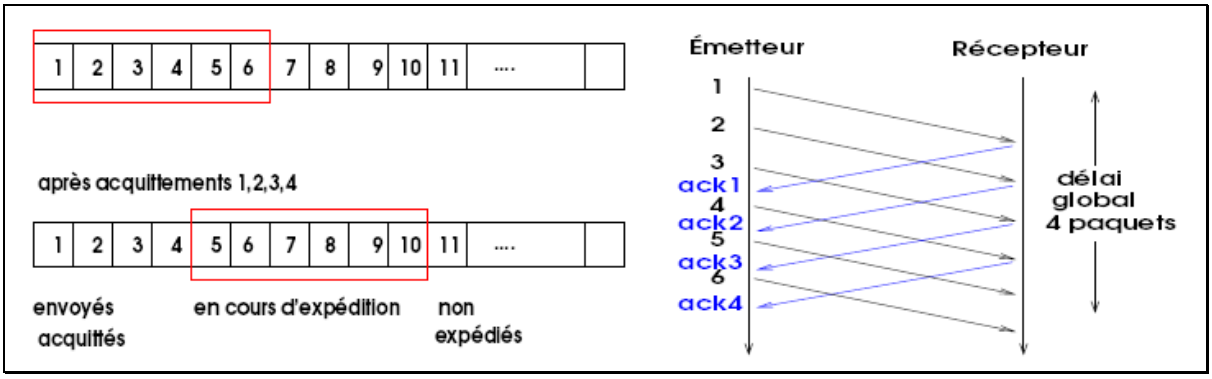
Tcp utilise un mécanisme consistant en l'´echange de trois messages pour la mise en place du circuit virtuel, connu comme une poignée de main tri directionnelle (three-way handshake)



SYN et ACK sont des bits du paquet tcp. Les numéros de séquence sont initialisés aléatoirement afin d'éviter que des paquets en retard initialisés à 1 viennent perturber ce fonctionnement.

7.4.1.2 Notion de fenêtre glissante :

C'est une fenêtre de taille fixe, représentant le nombre maximal de paquets transmis non acquittes. Une fenêtre de taille identique est créée du cote récepteur.

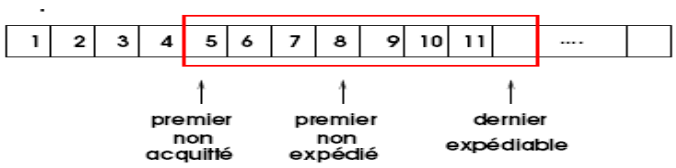


Le récepteur doit :

- enregistrer quels paquets n'ont pas été acquittes,
- conserver un temporisateur par paquet non acquitte.

7.4.1.3 Segments TCP et fenêtre glissante :

Puisque Tcp expédié des flots et ne s'intéresse qu'à la continuité de ces flots, on va considérer chaque suite d'octets expédiée dans un datagramme tcp comme un segment dans la suite à transmettre. Au lieu d'acquitter un paquet ou un segment, on acquitte une position :
acquitter la position n veut dire j'ai reçu tous les octets jusqu''a n. On en déduit que la fenêtre glissante est sur une limite d'octet et non de segment ou paquet. L'entité tcp d'expédition conserve trois pointeurs représentant l''etat de la fenêtre :



7.4.1.4 Taille de la fenêtre :

Tcp permet de modifier dynamiquement la taille de la fenêtre. Ainsi, on peut accroitre ou baisser le rythme d'expédition des données. En effet, la disponibilité (bande passante) courante du réseau et la

consommation de l'application réceptrice influent sur cette taille. On dit qu'on fait du contrôle de flux d'une Facon générale, lorsqu'on gère de Facon dynamique le rythme d'expédition (donc aussi de réception) des données entre deux entités. Comment fait Tcp ? Le récepteur indique dans chaque acquittement la taille de fenêtre acceptable pour lui, en fait, la taille courante disponible du tampon de réception.

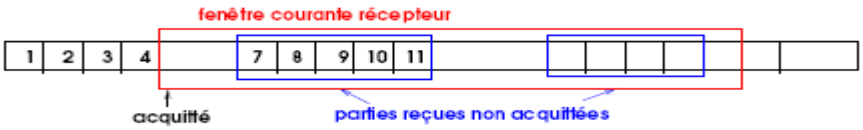
Remarques :

Il y a des acquittements qui annoncent uniquement une modification de la taille de la fenêtre. Une taille de zéro est possible. Cette Facon de faire permet à tcp d'agir indirectement sur les routeurs, dans des situations de congestion par exemple.

7.4.1.5 Point sur la situation :

On constate que tcp expédié les données dans des segments de taille variable et de plus, les segments retransmis peuvent être de taille différente des segments ayant provoqué la retransmission. Le récepteur peut donc reconstruire des parties pas forcement contigües.

Il n'empêche que l'acquittement ne doit se faire que sur la première partie Contigüe. En d'autres termes, le récepteur annonce toujours le prochain octet attendu.



INCONVENIENTS :

- La perte d'un acquittement est plus néfaste que celle d'un segment émis, car elle peut recouvrir plusieurs segments.
- L'´emetteur n'a aucune information sur ce qui a été reçu après la position acquittée.

Résume : grande fenêtre, gros risques, et réciproquement...

EXEMPLE :

Supposons que la position acquittée soit 1001. L'expéditeur a envoyé 4 segments de 1000 caractères chacun. Seul le premier s'est perdu. Apres l'´epuisement du délai d'acquittement, les schémas possibles sont tous inefficaces :

- retransmettre tous les 4 segments (taille de la fenêtre),
- ne retransmettre que le premier (suivre le standard).

En réalité, la seule chose qui peut sauver un peu cette situation est une annonce d'une nouvelle taille de fenêtre.

7.5 ANNEXE5 : LE PIXELGRABBER

```
public PixelGrabber(Image img,
                    int x,
                    int y,
                    int w,
                    int h,
                    int[] pix,
                    int off,
                    int scansize)
```

Create a PixelGrabber object to grab the (x, y, w, h) rectangular section of pixels from the specified image into the given array. The pixels are stored into the array in the default RGB ColorModel. The RGB data for pixel (i, j) where (i, j) is inside the rectangle (x, y, w, h) is stored in the array at pix[(j - y) * scansize + (i - x) + off].

Parameters:

img - the image to retrieve pixels from

x - the x coordinate of the upper left corner of the rectangle of pixels to retrieve from the image, relative to the default (unscaled) size of the image

y - the y coordinate of the upper left corner of the rectangle of pixels to retrieve from the image

w - the width of the rectangle of pixels to retrieve

h - the height of the rectangle of pixels to retrieve

pix - the array of integers which are to be used to hold the RGB pixels retrieved from the image

off - the offset into the array of where to store the first pixel

scansize - the distance from one row of pixels to the next in the array.

7.6 ANNEXE6 : LA REDIRECTION

7.6.1 REDIRECTION DIRECTEMENT SUR LE SERVEUR

Une configuration du serveur donne les règles de redirection. Il faut aller voir la documentation du serveur (Apache, IIS, etc.)

Pour le référencement : aucun problème. Vérifier tout de même l'entete HTTP renvoyé par la page (en utilisant par ou un outil d'analyse de l'entête HTTP ou directement mon outil de test de redirection).

7.6.2 REDIRECTION PAR URL REWRITING

La règle de redirection est indiquée dans un fichier .htaccess avec par exemple RedirectPermanent ou RewriteRule (dans ce cas il faut impérativement utiliser le code R=301)

Exemples : dans le fichier .htaccess situé à la racine du site (sur une seule ligne) :

`RedirectPermanent /robotstats http://www.robotstats.com/ ou bien (toujours sur une seule ligne) :`

RewriteRule ^article-([0-9]*).php archives-$1.htm [R=301]

Pour le référencement : fonctionne très bien avec tous les moteurs

7.6.3 REDIRECTION DANS UN SCRIPT SERVEUR (PHP, ASP, ETC.)

La redirection est définie par une fonction chargée de renvoyer un entête HTTP. Il faut bien sûr bien choisir le code de retour HTTP.

Exemple : utilisation de la fonction header() en PHP :

```
header("Status: 301 Moved Permanently", false, 301);
header("Location: http://www.votresite.com/unepage.htm");
exit();
```

Remarque : les deux derniers paramètres de la fonction header() sur la 1ère ligne de code ne sont pas toujours nécessaires. Mais sur certains serveurs, le code ci-dessous provoque une redirection 302 au lieu d'une redirection 301 :

```
header("Status: 301 Moved Permanently");
header("Location: http://www.votresite.com/unepage.htm");
exit();
```

Exemple : utilisation de la fonction addheader en ASP :

```
<%
response.status = "301 moved permanently"
response.addheader "location", "http://www.votre-site.com/"
response.end  %>
```

Pour le référencement : aucun problème.

7.6.4 REDIRECTION PAR BALISE META REFRESH

La redirection meta refresh est définie par la balise META http-equiv=”Refresh”. Elle donne l'ordre au navigateur de rediriger l'internaute vers une URL spécifiée au bout d'un certain nombre de secondes.

Exemple (redirection vers la page “nouvellepage.html” au bout de 5 secondes (sur une seule ligne) :

<meta http-equiv="Refresh" content="20;URL=page2.html">

Pour le référencement : il ne faut pas l'utiliser ! En effet, cette balise a trop souvent été utilisée en fixant le nombre de secondes à zéro, en général pour faire une page satellite. Même si Google liste parfois des redirections de ce type dans les backlinks, il faut la proscrire.

7.6.5 REDIRECTION JAVASCRIPT (OU TOUT AUTRE LANGAGE COTE CLIENT)

La redirection est définie par une fonction JavaScript qui modifie l'URL de la page à afficher, sans modifier l'entête HTTP.

Exemple :

```
<script language="javascript" type="text/javascript">
<!--  window.location.replace("http://www.un-site.com/une-page.htm");  -->
</script>
```

Pour le référencement : il ne faut pas l'utiliser ! Les robots ignorent le JavaScript, ils ne suivront donc pas une redirection JavaScript (il existe quelques exceptions).

7.6.6 LES LEADERS DANS LE DOMAINE

7.6.6.1 AXIS COMMUNICATIONS

Axis est une société informatique qui propose des solutions de vidéo sur IP à usage professionnel. Leader mondial du marché de la vidéo sur IP, Axis est à la tête de la transition actuelle de l'analogique vers le numérique en matière de vidéosurveillance. Centrés sur la surveillance et le contrôle à distance, les produits et solutions Axis reposent sur des plates-formes technologiques innovantes et ouvertes.

La position dominante d'Axis est le fruit de plus de vingt ans d'efforts fructueux visant à développer des technologies et des produits essentiels à la connectivité réseau, des canaux de distribution solides et des partenariats clés.

Avec plus de 1 million de produits professionnels de vidéo sur IP et plus de 3 millions de produits de mise en réseau installés, Axis sait assurément répondre à tous les besoins de ses clients.

Axis est reconnu comme une des marques les plus dignes de confiance sur le nouveau marché de la surveillance IP.

Quelques mots sur Axis:

- Société informatique dominant le marché de la vidéo sur IP.
- Fondée en 1984.
- Présence mondiale dans 20 pays, plus de 500 employés.
- Partenariats mondiaux avec des distributeurs, des revendeurs et des intégrateurs de systèmes dans plus de 70 pays.

7.6.6.2 ACTI

La Corporation ACTi est l'un des leaders de la technologie de surveillance IP, son activité principale est la sécurité et la surveillance des biens. La vaste innovation technologique d'ACTI est dans le développement de JPEG/MPEG-4/H.

La valeur essentielle d'ACTi est la compétence à développer. La technologie d'ACTi offre une solution complète permettant de couvrir tous les segments du marché de la sécurité. Non seulement l'offre de matériel de surveillance IP tels que les caméras IP et des serveurs vidéo, tous les produits d'ACTi sont livrés avec le logiciel de gestion libre ; en plus la riche sélection d'applications de gestion est offerts par les principaux fournisseurs de logiciels indépendants qui soutient les matériels d'ACTi.

7.6.6.3 ARTEC

Les technologies Artec développe, produit et distribue des logiciels supplémentaires, y compris les solutions basées sur des composants matériels, ainsi que des systèmes complets dans les domaines de la surveillance vidéo numérique et la technologie de la vidéo (Digital Vidéo Security) et de transmission numérique et des technologies d'enregistrement de contenus audio et vidéo pour l'intranet et la communication par Internet, par exemple, IPTV (Digital Vidéo).

Plus de 13.000 systèmes ont été livrés à ce jour là surtout ceux qui sont utilisés dans les domaines des services de sécurité.

Ces technologies mis au point une plate-forme qui permet l'intégration de différents matériels et les composants logiciels d'un vaste système de sécurité.

7.6.6.4 Architecture (matérielle et installation)

L'environnement du logiciel :

Le logiciel se présente sous la forme d'une interface web. Il sera donc imbriqué dans un serveur web lié à la base de données.

Interface matérielle :

Le logiciel s'intègre et s'adapte automatiquement au matériel sur lequel il tourne. L'utilisation d'une interface web comme support pour notre application permet une portabilité complète avec les différents systèmes d'exploitation présents sur le marché : Windows, Linux, Unix, Mac OS. La seule nécessité réside dans l'existence d'un réseau informatique LAN / WAN.

Le logiciel Rapace permet de diffuser très simplement et de manière complètement sécurisée le flux vidéo de webcams. Le flux de la caméra est alors accessible sur un compte personnel et sécurisé depuis le site Rapace.com.

Une fois la webcam installée, elle est directement reconnue par l'application Rapace. Celle-ci peut gérer jusqu'à 4 caméras simultanément. Et tant que l'application est en marche, on retrouve sur le compte web un accès à distance aux caméras.

Interface logicielle :

Il s'agit d'un logiciel qui ne s'interfacera pas avec d'autres logiciels.

Environnement opérationnel :

Le logiciel Rapace fonctionne via en amont un serveur central et un serveur local.

Installation:

Le service Rapace est basé sur l'utilisation d'un logiciel très simple qui vous permet de connecter votre Webcam et garder un œil sur ce qui vous est cher. Pour utiliser Rapace, voici les étapes à suivre :

- S'inscrire comme nouvel utilisateur
- Créer un compte sur le site
- Se connecter à son compte

On a alors accès en direct aux caméras se trouvant à l'endroit où le client les à installés. Le service Rapace fonctionne quelque soit le fournisseur d'accès internet

7.7 ANNEXE 8

7.7.1 QU'EST-CE QUE J2ME

Les principaux composants de la plate-forme Java 2, Micro Edition, (J2ME) incluent notamment des configurations CDC (Connected Device Configurations) et CLDC (Connected Limited Device Configurations), des profils MIDP (Mobile Information Device Profiles), ainsi que d'autres outils et technologies ayant trait aux solutions Java destinées au marché des appareils grand public et intégrés.

Les technologies J2ME incluent un environnement d'exécution Java hautement optimisé s'adressant spécialement à l'espace grand public. Les technologies J2ME sont adaptées à une vaste gamme de petits objets et offrent sécurité, connectivité ainsi que des programmes utilitaires très utiles aux cartes à puces intelligentes, aux pagers, décodeurs et autres petits appareils.

Les technologies J2ME ne sont qu'une partie de la famille de produits logiciels Java. Les plates-formes Java connexes sont les plates-formes Java 2, Standard Edition (plate-forme J2SE) et Java 2, Enterprise Edition (plate-forme J2EE). La technologie Java permet également de créer des services Web, de procéder à des transferts d'informations XML, de nombreux protocoles de réseau, des boîtes à outils, ainsi que l'application Java Web Start.

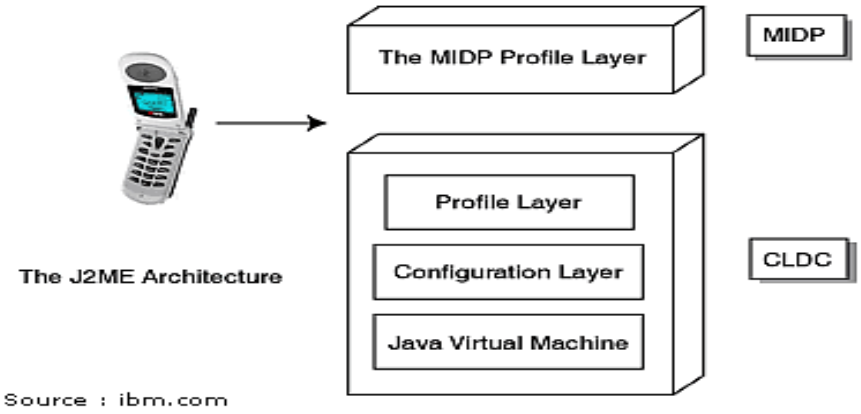
Il existe 2 configurations : Connected Device Configuration (CDC) et Connected Limited Device Configuration (CLDC) :

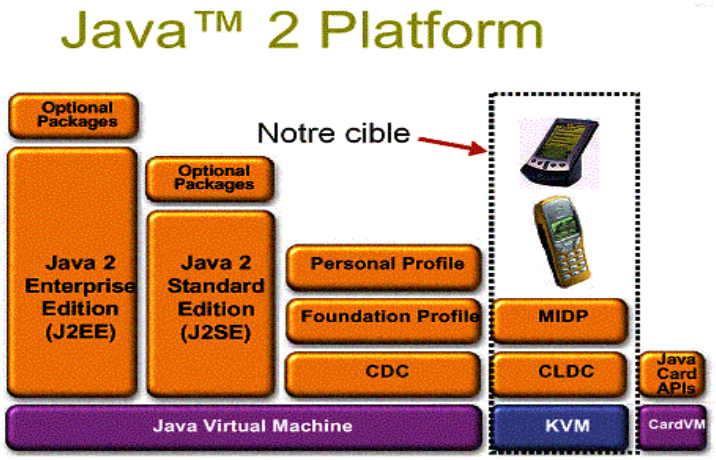
- **La CDC** : adaptée aux terminaux relativement puissant comme les PDA.
- **La CLDC :** dédiée aux appareils avec de faibles capacités comme les téléphones portables

7.7.2 CONNECTED LIMITED DEVICE CONFIGURATION

Le **Connected Limited Device Configuration** (CLDC) est un sous-ensemble des classes bibliothèques Java qui contient le minimum de programmes nécessaires pour faire fonctionner une machine virtuelle Java (JVM).

Le CLDC est essentiellement utilisé pour classer les multiples dispositifs dans une configuration fixe.





Une configuration offre les ensembles les plus élémentaires des bibliothèques et les caractéristiques de la machine virtuelle qui doivent être présents dans chaque mise en place d'un environnement J2ME.

Couplé avec un ou plusieurs profils, le Connected Limited Device Configuration donne aux développeurs une plate-forme solide Java avec laquelle on peut créer des applications pour les consommateurs et les systèmes embarqués.

L'API de base est définie par les sous-ensembles de Connected Limited Device Configuration.

7.7.3 API DE BASE

javax.microedition.io

Contient des classes Java ME spécifiques utilisés pour les opérations d'entrée-sortie.

javax.microedition.lcdui

"LCDUI interface utilisateur" tient compte du fait que les téléphones mobiles utilisent normalement des écrans LCD, mais les API ne sont pas spécifiquement adaptés à cette technologie d'affichage. On dit aussi que "LCD interface utilisateur" signifie "plus petit dénominateur commun» du fait des spécificités interface utilisateur à la conception la plus simple possible.

javax.microedition.rms

Record Management System (RMS), est à la fois une application et API pour le stockage sur les appareils J2ME, comme la cellule phones. Il fournit une forme de stockage persistant pour Java ME.

javax.microedition.midlet

Il contient les classes de base pour les applications Java ME.

7.7.4 API SPECIALISE AJOUTE AU MIDP (MOBILE INFORMATION DEVICE PROFILE)

javax.microedition.media

Il contient les classes de base de la lecture multimédia. Ce sont un sous-ensemble de la JSR 135 Java Mobile Media API.

javax.microedition.lcdui.game

Un jeu d'API visant à simple 2D sprite de jeux.

javax.microedition.pki

API d'authentification pour les connexions sécurisées.

javax.microedition.messaging

L'API de messagerie sans fil (en option), pour envoyer des SMS et des MMS. JSR120

javax.microedition.pim

La gestion des renseignements personnels API (facultatif), l'accès du périphérique carnet d'adresses.

javax.microedition.io.file

Le fichier de connexion en option Forfait (FCOP) est l'un des deux paquets facultatifs définis par JSR75 via le Java Community Process. Le FileConnection spécifié dans l'API JSR 75 donne accès à la boucle locale des systèmes de fichiers sur des appareils comme les PDA. Afin de surmonter les questions de sécurité MIDlet il doit inclure un fichier de demande d'autorisation dans son fichier JAD Midlet, sous-autorisation de propriété.