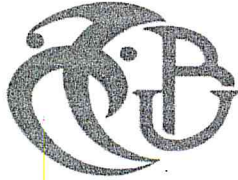


MH 19-1
République Algérienne Démocratique et Populaire
Ministère de l'enseignement Supérieur et de la recherche Scientifique

Université Saad Dahleb, Blida
USDB



Faculté des sciences
Département d'informatique



Mémoire de fin d'étude

Pour l'obtention du diplôme Master en informatique

Spécialité : Ingénierie des logiciels

Sujet :

Conception d'une application web pour la
gestion des alertes relatives aux
équipements VSAT

Présenté par :

ELHORR Yasmine
FOURALI Manel

Promoteur :

Mr SIDOUMOU Redha

Organisme d'accueil : ATS

Soutenu le : 14/10/2010

Président : Melle Aoussat

Rapporteur : Mr Sidoumou

Examineur : Melle Elghers, Mr Bala

, devant le jury composé de



Dédicace

Je dédie ce modeste travail

- ▲ la mémoire de mes deux grand-pères ;
- ▲ mes parent, ces deux êtres très chers, pour leur soucis de me voir réussir et en témoignage de mon amour et ma reconnaissance ;
- ▲ mes chers frères : Abd el alim, Mouloud, Akram qui m'ont toujours soutenu et encouragée, qu'ils en soient remerciés ;
- ▲ mes deux grand-mères pour leurs prières, et particulièrement à ma grand-mère paternelle qui m'a accueilli chez elle durant mon cursus universitaire;
- ▲ tous les membres des familles Fourali et Ouchenir;
- ▲ la sœur que je n'ai jamais eu : Hiba , pour avoir toujours répondu présente dans les moments difficiles;
- ▲ mes chers (es) amis (es) avec qui j'ai eu le plaisir de partager mes années d'études, pour tous ces instants de pur bonheur que vous me procuriez et en témoignage de mon amitié sincère;
- E**t surtout a ma très chère amie et binôme Yasmine pour toute la bonne humeur qu'elle diffuse autour d'elle.
- ▲ tous ceux qui ont cru en moi, et qui m'ont motivée, qu'ils trouvent ici l'expression de mon amour et ma profonde gratitude;



Manel





Dédicace

Je dédie ce modeste travail

▲ mes parent, ces deux êtres très chers, pour leur soucis de me voir réussir et
en témoignage de mon amour et ma reconnaissance ;

▲ ma chère sœur Nadia et mon cher petit frère Mohamed ;

▲ Hssn que dieu le garde pour moi ;

▲ tous les membres de ma grande famille maternelle et paternelle ;

▲ tous mes amis (es) avec qui j'ai eu le plaisir de partager mes années d'études,
pour tous ces instants de pur bonheur que vous me
procuriez et en témoignage de mon amitié sincère;

Et surtout à ma très chère amie et binôme Manel pour toute la bonne humeur qu'elle
diffuse autour d'elle ;

▲ tous ceux qui ont cru en moi, et qui m'ont motivée, qu'ils trouvent ici l'expression
de mon amour et ma profonde gratitude.



Yasmine



Remerciements

Au terme de ce travail, nous tenons à exprimer notre profonde gratitude à notre encadreur *Mr Amirouche Fateh* d'Algérie Télécom Satellite qui a accepté de nous octroyer ce stage au sein de l'entreprise.

Nous le remercions également pour sa patience à notre égard et sa disponibilité durant toute la période de stage.

Nos vifs remerciements vont également à *Mr Sidoumou* qui a accepté d'être notre promoteur. Et aussi pour l'aide et le temps qu'il a consacrés à la mise en ordre des différentes idées contenu dans ce rapport.

Nous voudrions remercier aussi les membres du jury d'avoir accepté de juger notre travail.

Que le corps professoral et administratif du département d'informatique trouve ici nos vifs remerciements.

Nos remerciements s'adressent aussi à tout le personnel d'Algérie Télécom Satellite pour leur accueil.

Nous remercions enfin toute personne qui a contribué de près ou de loin à l'élaboration de ce travail.

Table des matières

<i>Introduction générale</i>	9
<i>Chapitre I: Les satellites et la technologie VSAT</i>	10
1. <i>Introduction</i>	10
2. <i>Les satellites</i>	10
2.1. <i>Historique</i>	10
2.2. <i>Notion sur les satellites</i>	11
2.2.1. <i>Définition de satellite</i>	11
2.2.2. <i>Les orbites</i>	12
2.2.3. <i>Les bandes de fréquences</i>	13
2.3. <i>Les contraintes des solutions satellites</i>	14
2.3.1. <i>La couverture</i>	14
2.3.2. <i>La gestion de la bande passante</i>	15
2.3.3. <i>Le délai</i>	16
3. <i>Présentation de la technologie VSAT</i>	17
3.1. <i>Organisation du système satellite</i>	17
3.2. <i>Architectures des réseaux VSAT</i>	18
3.2.1. <i>Architecture maillée</i>	18
3.2.2. <i>Architecture en étoile</i>	19
3.2.3. <i>Architecture mixte</i>	20
3.3. <i>Gestion de la bande passante</i>	20
3.4. <i>Les solutions VSAT</i>	21
3.4.1. <i>DVB/RCS</i>	21
3.4.2. <i>SCPC</i>	22
3.5. <i>Les applications</i>	23
4. <i>Pourquoi choisir cette technologie</i>	23
4.1. <i>Quand penser à VSAT</i>	23
4.2. <i>Les avantages du VSAT</i>	24
4.3. <i>Les inconvénients</i>	24
5. <i>Conclusion</i>	25
5.1. <i>L'importance de l'étude préliminaire</i>	25

5.2. <i>Ou en est le VSAT</i>	25
Chapitre II: Généralités sur la gestion de réseau	26
1. <i>Introduction</i>	26
2. <i>Les disciplines de l'administration d'un système</i>	26
2.1. <i>L'administration des utilisateurs</i>	26
2.2. <i>L'administration des serveurs</i>	27
2.3. <i>L'administration de la machine de transport</i>	27
3. <i>Fonctionnalités de la gestion de réseau</i>	28
3.1. <i>La cueillette des informations de gestion</i>	28
3.2. <i>L'interprétation des données</i>	29
3.3. <i>Le contrôle des équipements</i>	29
4. <i>Principes de la gestion de réseau</i>	29
5. <i>Le protocole SNMP</i>	30
5.1. <i>Composants du protocole SNMP</i>	30
5.1.1. <i>Le managerSNMP</i>	31
5.1.2. <i>L'agent SNMP</i>	32
5.1.3. <i>La MIB</i>	33
5.2. <i>Fonctionnement du protocole SNMP</i>	35
5.2.1. <i>Les requêtes SNMP</i>	35
6. <i>Les outils de supervision réseau</i>	36
6.1. <i>Outis SNMP</i>	36
6.2. <i>MRTG</i>	37
6.3. <i>RRDTOOL</i>	37
7. <i>Conclusion</i>	39
Chapitre III: Méthode d'analyse et de conception	40
1. <i>Introduction</i>	40
2. <i>Le langage UML</i>	40
3. <i>Les processus de développement logiciel</i>	41

3.1. Processus unifié	41
3.2. Le processus 2TUP	42
3.2.1. Définition	42
3.2.2. Branche fonctionnelle	43
3.2.3. Branche technique	44
3.2.4. Branche conception / réalisation	44
4. Conclusion	45
Chapitre IV : Etude préliminaire	46
1. Introduction	46
2. Présentation du projet	46
3. Recueil des besoins fonctionnels et non fonctionnels	47
a. Exigences fonctionnels	47
b. Exigences non fonctionnels	47
4. Identification des acteurs du systèmes	48
5. Identification des messages	48
6. Modélisation du contexte	49
7. Conclusion	50
Chapitre V: Etude fonctionnelle	51
1. Introduction	51
2. Capture des besoins fonctionnels	51
2.1. Identification des cas d'utilisation	51
2.2. Les diagrammes des cas d'utilisation	52
3. Analyse	59
3.1. Développement du modèle statique	59
3.2. Développement du modèle dynamique	64
4. Conclusion	73
Chapitre VI : Etude technique	74
1. Introduction	74
2. Captures des besoins techniques	74

2.1. Spécification d'architecture	74
2.2. Elaboration du modèle de spécification logicielle	77
3. Conception générique	79
3.1. Les grands choix techniques	80
3.1.1. Langage de programmation	80
3.1.2. Serveur d'application	84
3.1.3. Système de gestion de base de données(SGBD)	85
3.1.4. Environnement de développement	86
4. Conclusion	87
Chapitre VII : Conception et réalisation	88
1. Introduction	88
2. Conception préliminaire	88
2.1. Enumération des interfaces utilisateur	88
3. Conception détaillée	90
3.1. Conception des classes	90
3.2. Conception des attributs	90
3.3. Règle de passage du modèle de conception au modèle relationnel	91
3.1.1. Transformation des classes (règle1)	91
3.1.2. Transformation des relations	91
3.1.2.1. Relation un-à-plusieurs (règle2)	91
3.1.2.2. Relation plusieurs-à-plusieurs (règle3)	92
3.1.2.3. Relation un-à-un (règle4)	93
3.1.2.4. Relation d'héritage (règle5)	93
3.1.2.5. Relation de composition (règle6)	94
3.1.3. Schéma relationnel de la base de données	95

<i>4.Réalisation</i>	96
4.1. <i>Page d'accueil</i>	96
4.2. <i>Liste des clients, équipements et composants</i>	97
4.3. <i>Liste des alertes</i>	97
4.5. <i>Détail d'une alerte</i>	98
4.6. <i>Détails d'un composant</i>	99
4.7. <i>Chargement de l'historique des alertes</i>	99
4.8. <i>Envoi d'un rapport d'alerte</i>	100
 <i>Conclusion générale</i>	 101
 <i>Bibliographie</i>

Liste des figures

Figure I.1. Fréquences des services satellites dans la région	14
Figure I.2. Illustration d'un réseaux VSAT.....	18
Figure I.3. Architecture d'un réseau maillé.....	19
Figure I.4. Architecture d'un réseau en étoile.....	19
Figure I.5. Illustration de la division d'un segment spatial en cannaux	20
Figure I.6. Les applications d'un réseau VSAT.....	23
Figure II.1. Architecture du protocole SNMP.....	31
Figure II.2. Structure d'indexation des données dans la MIB SNMP	31
Figure II.3. Génération de graphiques dans MRTG.....	37
Figure II.4. Génération de graphiques dans RRDTOOL.....	38
Figure III.1. Les contraintes soumises aux systèmes.....	43
Figure III.2. Le processus de développement en Y.....	43
Figure IV.1. Diagramme de contexte du système de gestion des alertes	52
Figure V.1. Diagramme de cas d'utilisations « accès au système ».....	53
Figure V.2. Diagramme de cas d'utilisations « Processus de gestion des alertes ».....	54
Figure V.3. Diagramme de classe du système de gestion des alertes.....	60
Figure V.4. Diagramme de séquences du cas d'utilisation « accès au système ».....	65
Figure V.5. Diagramme de séquences du cas d'utilisation « consulter les alertes».....	66
Figure V.6. Diagramme de séquences du cas d'utilisation « consulter les détails d'un composant ».....	67
Figure V.7. Diagramme de séquences du cas d'utilisation « consulter l'historique des alertes».....	68
Figure V.8. Diagramme de séquences du cas d'utilisation « envoyer un rapport d'alerte»	69
Figure V.9. Diagramme de séquences du cas d'utilisation « générer les alertes ».....	70
Figure VI.1. Architecture 3 tiers.....	76
Figure VI.2. Diagramme des cas d'utilisations technique.....	77
Figure VI.3. Diagramme de déploiement du système de gestion des alertes.....	87
Figure VII.1. Transformation d'une classe.....	90
Figure VII.3. Transformation d'une relation plusieurs-à-plusieurs.....	92
Figure VII.4. Transformation d'une relation un-à-un.....	93

Figure VII.5. Transformation d'une relation d'héritage.....	94
Figure VII.6. Transformation d'une relation de composition.....	94
Figure VII.7. Schéma relationnel de la base de données.....	95
Figure VII.8. Fenetre de page d'accueil.....	96
Figure VII.9. Fenêtre de la liste des clients, équipements et leurs composants.....	97
Figure VII.10. Fenêtre de la liste des alertes d'un composant.....	97
Figure VII.11. Fenêtre de la liste des alertes d'un équipement.....	98
Figure VII.12. Fenêtre des détails d'une alerte.....	98
Figure VII.13. Fenêtre des détails d'un composant.....	99
Figure VII.14. Fenêtre de chargement de l'historique des alertes.....	100
Figure VII.15. Fenêtre de l'envoi d'un rapport d'alerte.....	100

Liste des tableaux

Tableau III.1. Synthèse des méthodologies utilisées dans le cadre de développement objet et nouvelles technologies.....	42
Tableau IV.2. Les messages émis par le système.....	49
Tableau IV.3. Messages reçus par le système.....	50
Tableau V.4.Cas d'utilisation du système de gestion des alertes.....	58
Tableau V.5.Tableau de descriptions des classes du système de gestion des alertes.....	58
Tableau V.6. Tableau de correspondance entre modèle statique et dynamique.....	58
Tableau VI.7. Tableau descriptive du diagramme des cas d'utilisation techniques	78
Tableau VII.8. Tableau des vue IHM	89

Résumé :

Dans cet espace technologique en continuel développement il devient nécessaire de choisir une méthode de connexion au réseau mondial de la communication qui est Internet par des moyens fiables basés sur des technologies de pointe. Dans cette optique *ATS* (Algérie Télécom Satellite) met à la disposition de ses clients une solution *VSAT* (*Very Small Aperture Terminal*) offrant des services de communication par satellites plus fiable que celles offertes par les réseaux filaires.

De ce fait, le besoin de surveiller les équipements VSAT mis à la disposition de ses clients et d'agir quand une erreur se produit devient grandissant. Or, le montant des investissements, parfois prohibitif, dans un système de management de réseaux (network management system, NMS) freine l'emploi de cet outil.

Dans le cadre du monde de l'Open source, les solutions sont nombreuses mais ne correspondent pas parfaitement aux exigences d'*ATS*, l'inconvénient majeur de ses solutions est qu'elles ne prennent en charge aucune configuration des évènements qui avertissent l'administrateur du réseau d'un dysfonctionnement d'un équipement ou activité anormale au sein du réseau.

Notre travail consiste en l'élaboration d'une application web pour la gestion des alertes relatives aux équipements *VSAT*, afin de surveiller l'état des équipements en permanence et assurer une meilleure qualité de services.

Abstract :

It has become necessary in a continuously developing technology environment to choose reliable ways to access a global network; the Internet. In this view, *ATS* (Algérie Télécom Satellite) provides its clients with a *VSAT* (*Very Small Aperture Terminal*) communication solution that relies on satellites, positioning itself as a cutting edge solution as compared to a wire network.

This connectivity scenario raises the need to monitor VSAT equipment put at the user's disposal, and act upon malfunctions, through a network management system (NMS). However, the costs associated with this system hinder its implementation.

Many alternative solutions exist within the open source framework, but these do not necessarily meet *ATS*'s requirements; their main inconvenience being the lack of a triggering system that warns the network administrator of any network failures.

Our project consists of putting in place a web application for the management of events threatening the good operation of VSAT equipment, continuously monitoring the state of the system and ensuring a high quality of service.

Introduction générale

1. Introduction

Les réseaux informatiques ont aujourd'hui beaucoup d'importance, au point que la plupart de nos activités ne pourraient plus être envisagées sans la mise en place de ces réseaux. On assiste à leur déploiement à tous les niveaux de la société, dans les entreprises, au niveau national et international, y compris dans les domiciles des usagers.

Le réseau constitue l'environnement virtuel d'interactions entre les individus, les applications et les ordinateurs. Les activités des entreprises deviennent aussi dépendantes du fonctionnement de ce moyen de communication que de celui de leur système d'information.

Pouvoir étendre rapidement son réseau et connecter différents sites fixes et mobiles, disposer d'une tarification réaliste par rapport aux besoins, nécessité de solution unique pour transporter data, voix, vidéo, IP...., saturation des réseaux terrestres, répondre à des applications de communication d'entreprise, telles sont les nouvelles contraintes auxquelles doivent faire face les responsables des systèmes d'information. Ces contraintes ont motivé l'apport d'une nouvelle solution de communication à savoir la communication par satellites basée sur la technologie *VSAT (Very Small Aperture Terminal)* qui permet une extension du réseau notamment vers les sites pauvres en infrastructure de télécommunication terrestres. Les avantages de ce type de réseau ne sont plus à démontrer.

Le développement des réseaux ne s'effectue pas sans contrepartie. En effet, tant les activités des entreprises que les individus eux-mêmes sont de plus en plus dépendants de ces réseaux. Aussi est-il donc crucial que les services de communications du réseau soient disponibles en permanence. Les éventuels dysfonctionnements ou pannes doivent être détectés le plus rapidement possible et traités dans un délai compatible avec les activités de l'entreprise.

Ainsi de nombreuses solutions de gestion de réseau open sources ont été proposées, dont la majorité est fondée sur le protocole *SNMP (Simple Network Management Protocol)*. Comme son nom l'indique, il s'agit d'un protocole simple qui permet aux administrateurs réseau de gérer les équipements IP du réseau et de diagnostiquer les problèmes

Chapitre I: Les satellites et la technologie VSAT

1. Introduction

La communication de données par satellite permet de relier par radio deux points distants de plusieurs centaines de kilomètres. Le principal avantage du satellite par rapport aux autres technologies est qu'il couvre de très grandes étendues. En revanche, les performances sont plus limitées qu'avec les technologies terrestres, en particulier filaires (latence élevée, débits limités et partagés), ainsi que le coût qui peut représenter un obstacle pour l'utilisateur. Face à la problématique des zones blanches durables de l'ADSL, la connexion par satellites constitue l'une des solutions envisageables, c'est sur ce dernier segment qu'il se développe aujourd'hui.

2. Les satellites

2.1. Historique

La conquête de l'espace est une passion qui a depuis longtemps fasciné l'esprit des savants. Par contre l'idée de placer un objet en orbite autour de la terre a commencé à germer seulement au début du vingtième siècle. Il a fallu attendre 1945 pour que le concept de satellite voie le jour. C'est le britannique *Arthur C. Clarke* qui introduit le premier, le concept de communication par satellite.

Le 4 octobre 1957, l'Union Soviétique lance le premier satellite artificiel : Spoutnik. Ce satellite était une simple sphère métallique d'un diamètre de 58 cm équipée d'un simple émetteur radio [Web 1].

✂ L'évolution des satellites

Les premiers satellites furent d'abord passifs ; ils se contentaient simplement de réfléchir les signaux émis par les stations terrestres. L'inconvénient majeur était que ces signaux étaient diffusés dans toutes les directions et pouvaient donc être captés dans n'importe quelle partie du monde. De plus, il fallait être équipé de grandes antennes au sol pour pouvoir disposer d'une grande puissance d'émission et de réception. En 1960, les Américains mettent en orbite leur premier satellite en mode passif : *Echo 1*.

Les deuxièmes types de satellite furent ensuite actifs. C'est-à-dire qu'ils possédaient leur propre système de réception et d'émission. Le premier satellite actif, *Telstar 1*, fut américain.

Mis en orbite deux ans après *Echo 1*, ce satellite disposait d'un enregistreur à bande qui enregistrerait les données lors de son passage au dessus d'une station émettrice.

Ensuite, il les diffusait lorsqu'il se situait au-dessus d'une station réceptrice. L'exploitation commerciale de ce type de satellite ne commença qu'en 1965.

§ Les satellites commerciaux

Pour contrer le frein que représentait le coût de cette technologie à l'époque, INTELSAT (*International Telecommunications Satellite Consortium*) vu le jour en 1964. Cette société était une coopération entre états. Elle conquit les deux tiers des communications intercontinentales et divisa par huit le coût de location de la capacité de transmission embarquée sur un satellite.

Mais il a fallu attendre les années 80, pour assister au « boum des satellites commerciaux ». Cette explosion a pu se faire grâce à la déréglementation de ce secteur. Le détonateur a pour nom *Livre vert* sur les communications par satellites. Ce texte proposa des mesures de libéralisation du secteur terrestre sous réserve de procédure d'homologation et le libre accès au secteur spatial sous réserve de licence.

Aujourd'hui, il y a dans le ciel plus d'une centaine de satellites qui diffusent constamment de la téléphonie, de la télévision, des data... Ce nombre commence à poser certains problèmes au niveau des bandes de fréquences utilisées qui commencent à être quelque peu saturées.

2.2. Notions sur les satellites [Web2]

Avant de s'intéresser aux technologies satellites et VSAT en particulier, il est bon de rappeler quelques notions sur les satellites, leurs particularités et les contraintes de cette technologie. Tout d'abord précisons ce qu'est un satellite.

2.2.1. Définition de satellite

Un satellite de télécommunication peut être considéré comme une sorte de relais hertzien. En effet, il ne s'occupe pas de la compréhension des données : ce n'est qu'un simple miroir. Son rôle est de régénérer le signal qu'il a reçu et de le retransmettre amplifié en fréquence à la station réceptrice. Le satellite offre également une capacité de diffusion, c'est-à-dire qu'il peut retransmettre les signaux captés depuis la terre vers plusieurs stations. La démarche inverse peut également être effectuée ; il peut récolter des informations venant de plusieurs stations différentes et les retransmettre vers une station particulière. De plus, il est également possible

d'établir des liaisons directes entre satellites. Ce principe a d'ailleurs été utilisé dans le projet *Iridium* développé par *Motorola*.

Donc pour résumer on peut dire qu'un satellite est un élément spatial qui a pour rôle de produire ou relayer des données vers différents récepteurs terrestres.

L'avantage évident présenté par les solutions satellites est que les stations terrestres ne dépendent plus des infrastructures terrestres existantes à travers le monde et donc peuvent être mobiles.

Par ailleurs les transmissions satellites permettent de mettre en œuvre aisément (à comparer avec les structures câblées) les principes de diffusion. En effet il est possible de diffuser facilement et de façon économique (en bande) depuis un satellite la même information à de nombreuses stations ou à l'inverse relayer depuis un satellite la synthèse de multiples sources terrestres ou spatiales.

2.2.2. Les orbites

Les satellites utilisent la force gravitationnelle de notre planète afin de se maintenir à une position et à une distance déterminée de la terre. Il est ainsi possible de définir à tout moment quelles sont les caractéristiques du satellite pour établir des transmissions. Nous allons voir dans cette partie quelles sont les types d'orbites utilisées et comment celles ci fixent certaines limites ou contraintes dans les transmissions ou les équipements.

☞ L'orbite géostationnaire

L'orbite géostationnaire est sans aucun doute l'orbite la plus répandue aujourd'hui, car le satellite se déplace en même temps que la terre ; il fait donc le tour de la terre en 24h (durée qui correspond au temps de rotation de la terre) et paraît donc immobile dans le ciel. Dans cette orbite, le satellite est placé à 35 786 Km d'altitude et peut couvrir une large calotte d'une superficie qui peut atteindre un hémisphère.

L'avantage de ce type de satellites, est que l'on peut utiliser des antennes fixes au sol. Ils présentent cependant l'inconvénient d'être situés bas sur l'horizon lorsqu'ils couvrent des zones éloignées se rapprochant des pôles: les signaux à transmettre parcourant une plus grande distance, subissent des atténuations plus importantes, et mettent plus de temps à arriver sur terre (un quart de seconde en plus).

⌘ **Orbite circulaire polaire**

Comme leur nom l'indique, ces satellites passent au-dessus des deux pôles, et peuvent au bout d'un certain temps couvrir toute la surface du globe. Leur domaine d'application se situe surtout dans l'observation ou la communication différée.

Le satellite d'observation français *Spot*, situé à 800 km d'altitude, permet de couvrir toute la surface du globe en 21 jours.

⌘ **Orbite circulaire incliné**

Ce type de satellite ne permet pas de couvrir en totalité toute la surface du globe, puisque la plus haute altitude desservie correspond à l'inclinaison du plan orbital.

Le projet *GlobalStar*, réunissant les entreprises *France Télécom*, *Alcatel*, *Qualcomm* et *Laural*, a lancé 48 satellites en orbite circulaire inclinée à 50° par rapport à l'équateur, dans le but d'assurer des communications mobiles (à l'usage de personnes en déplacement) depuis l'essentiel de la surface terrestre. La puissance des antennes omnidirectionnelles des mobiles étant limitée (de l'ordre de 1 W), il faut donc jouer sur la capacité des antennes des satellites. Comme plus un satellite est éloigné de la Terre, plus grande doit être son antenne, les satellites du projet *GlobalStar* ont donc été placés sur des orbites basses.

⌘ **Orbite elliptique**

Les satellites en orbite elliptique ont une vitesse très variable en fonction de l'endroit où ils se placent sur l'ellipse. Ils n'occupent donc pas une position fixe par rapport à la terre, ce qui suppose d'utiliser des antennes terrestres mobiles pour suivre ces satellites, contrairement aux satellites géostationnaires. Par contre, ils possèdent l'avantage de pouvoir desservir plus aisément des zones éloignées de l'équateur sous un angle assez élevé, ce qui implique que les signaux à transmettre traversent une couche atmosphérique plus étroite.

Ce type de satellite est utilisé depuis 1967 par le système *Molnya* qui assure des télécommunications pour la Sibérie.

2.2.3. Les bandes de fréquences

Pour éviter un chaos total dans le ciel, une réglementation internationale spécifique et stricte a été mise en place par l'*Union Internationale des Télécommunications* (UIT-T) concernant la répartition des fréquences; elle fait partie intégrante du règlement international des radiocommunications. Cette réglementation définit notamment la position orbitale des satellites et les bandes de fréquences qu'ils doivent utiliser et respecter. Plusieurs types de services de communications par satellites sont définis dans la réglementation : le service fixe

par satellite (SFS), le service mobile par satellite (SMS), qui comporte un service mobile terrestre et un service mobile maritime, le service de radiodiffusion par satellite (SRS).

Fréquences des services satellites dans la région 1 (Europe, Afrique et Asie du Nord)		
Bande	Sens montant/descendant	Largeur de gamme
Services fixes par satellite		
Bande C	6/4 GHz	1100 MHz
Bande X	8/7 GHz	500 MHz
Bande Ku	14/11 GHz	1000 MHz
Bande Ku	14/12 GHz	250 MHz
Bande Ka	30/20 GHz	2500 MHz
Services mobiles par satellite		
Bande L	1,6/1,5 GHz	29 MHz
Services de radiodiffusion par satellite		
Bande K	17/12 GHz	800 MHz

Figure I.1 : Fréquences des services satellites dans la région 1

Il existe également une répartition géographique en 3 régions :

- ✓ La région 1 (Europe, Afrique, Moyen-Orient et l'Union Soviétique).
- ✓ La région 2 (Asie, Océanie)
- ✓ La région 3 (Amérique)

2.3. Les contraintes des solutions satellites

Comme nous l'avons précédemment vu les satellites sont largement définis par leur orbite qui fixe des contraintes particulières (types de matériels, gestion de la position du satellite) pour la gestion des transmissions .Nous allons maintenant nous intéresser à d'autres contraintes qui sont aussi liées aux orbites mais qui peuvent être contournées par l'implantation de gestions ou d'architectures adaptées. Le choix final sera fixé par le rapport entre le coût et le respect des besoins.

2.3.1. La couverture

L'orbite d'un satellite de par sa forme et son rayon définit la zone de couverture et la portée du satellite. Plus le satellite est éloigné de la terre et plus sa couverture est étendue. Bien qu'évident ce critère reste un élément majeur dans le choix et l'élaboration d'une solution satellite. En effet plusieurs systèmes de satellites peuvent couvrir la même superficie mais chacun se distinguera par un ensemble de caractéristiques particulières. Parmi elles, la

plus déterminante est le nombre de satellites composant le système et la méthode utilisée pour les gérer.

En effet, la couverture d'un satellite géostationnaire peut être atteinte par une constellation de satellites à plus basse altitude mais il faudra alors s'intéresser aux moyens de rendre ce réseau homogène sur l'ensemble de la zone. Pour cela deux solutions existent soit les satellites communiquent entre eux soit un relais terrestres permet de les synchroniser.

Ce choix doit aussi tenir compte du nombre de stations terrestres, de leur densité et des évolutions futures. Une société multinationale avec de gros débit aura peut être plus intérêt à investir et privilégié une solution géostationnaire pour une couverture vaste plutôt que de favoriser une zone du monde avec une solution autre.

2.3.2. La gestion de la bande passante

Pour diffuser les données, qu'elles soient numériques ou analogiques, les stations terrestres accèdent aux satellites par l'intermédiaire de fréquences spécifiques. En effet l'acquisition d'un support de transmission satellite est en fait la location d'une bande de fréquences qui sera consacrée et partagées par les différentes stations de ce réseau satellite.

Sans politique d'accès pour accéder au support, les signaux transmis par une station se confondraient avec d'autres signaux provenant de stations différentes. Les signaux reçus seraient alors incompréhensibles et impossibles à décoder ; cela entraînerait leur perte et il serait nécessaire de les retransmettre. De plus, il n'est pas envisageable d'allouer un canal pour chaque station ; ce système serait beaucoup trop coûteux. La mise en place d'une politique d'accès aux canaux satellites a donc été réalisée pour dans un premier temps, permettre à plusieurs stations d'accéder à un même canal de transmission, et dans un deuxième temps, pour avoir une exploitation maximale des transpondeurs du satellite tout en garantissant qu'il y ait le moins de collisions possibles. (Il est à garder à l'esprit qu'une solution satellite demande un fort investissement, ce médium doit donc être optimisé au maximum)

Ce partage de la bande passante est aussi soumis à certaines prérogatives liées aux applications, aux particularités intrinsèques des types des satellites et à leur nombre.

Le cas le plus simple est celui du satellite géostationnaire seul .En effet le partage de la bande est réalisé ici de façon unique et les calculs pour la répartition des canaux ne tient pas en compte les baisses de puissances dues aux déplacements du satellite par rapport aux stations. En effet une station utilisera toujours le même satellite et son antenne aura une position fixe.

Le hand over intra satellite :

Il correspond à une réattribution de canal pour une ou plusieurs stations au sein du même satellite. Cela est réalisé pour optimiser les échanges lorsqu'un canal est peu utilisé ou très perturbé par exemple, cette technique peut aussi être utilisée pour la répartition de charge.

Le hand over inter satellite :

Cette situation est directement liée à la mobilité du satellite ou des stations.

Le changement de canal pour la ou les stations est effectué dans ce cas lorsqu'une transmission est basculée sur un autre satellite.

Pour réaliser ces attributions de canaux on peut distinguer plusieurs approches :

- ✂ **Le Soft Hand over** « mou » dans ce cas le basculement d'un premier canal vers un second (sur le même satellite ou non) passe par un état de transition où la transmission est maintenue sur les deux canaux avant de se fixer sur le nouveau.
- ✂ **Le Hard hand over** « dur » quant à lui fait basculer instantanément la transmission d'un canal à l'autre. Ces techniques dépendent des contraintes de temps et d'intégrité de nos besoins.

Une autre approche permet d'anticiper les basculements (en définissant des zones ou des seuils critiques) ou de réserver des canaux pour gérer ces hand over. [Web3]

2.3.3. Le délai

Le délai d'un système par satellite géostationnaire est d'environ 270 millisecondes : c'est le temps que prend un signal pour parcourir 35 800 Km dans l'espace et revenir. En ajoutant à cette durée le temps requis pour le traitement des signaux par le matériel du satellite et de la bande de base, on obtient un délai total de près de 320 millisecondes.

Certaines applications de par leur nature (les applications temps réel par exemple) ne pourront donc pas être supportées par toutes les structures satellites ou du moins pas avec les mêmes performances.

Le délai de propagation ne cause pas de problèmes insurmontables dans la conception et l'exploitation d'un réseau informatique par satellite, sauf dans les cas où l'on se contente de substituer un support de transmission à un autre.

D'autant plus que ce délai, même s'il paraît important au vu d'autres technologies, peut être tout à fait acceptable pour certaines applications, pour les transferts de données par exemple qui privilégient la fiabilité et les débits.

La réponse aux problèmes occasionnés par le délai au sein des réseaux informatiques par satellite est fournie par l'emploi de protocoles perfectionnés ou de compensateurs de temps de

propagation qui envoient un accusé de réception à l'échelle locale avant la transmission des données par satellite, ce qui élimine le retard dans la prise de contact des protocoles. La nouvelle génération des stations terrestres à très petite ouverture d'antenne (*VSAT*) et certains multiplexeurs comportent des compensateurs de délai et des convertisseurs de protocoles, appelés assembleurs désassembleurs de paquets, qui assurent l'établissement de la liaison à l'échelle locale et modifient les protocoles pour répondre aux exigences du satellite.

3. Présentation de la technologie VSAT

Avant de commencer la présentation de ce système, il faut savoir que le *VSAT* n'est pas une technologie normalisée mais plutôt un concept.

En effet, chaque constructeur a sa propre manière d'implémenter le système. Même si tous les systèmes fonctionnent sur le même principe, la plupart des détails techniques et des définitions de protocoles utilisés sont bien gardés par chaque constructeur.

3.1. Organisation du système satellite

Le *VSAT* (Very Small Aperture Terminal) est un système qui repose sur le principe d'un site principal (le hub) et d'une multitude de points distants (les stations *VSAT*).

Le hub est le point le plus important du réseau, c'est par lui que transite toutes les données qui circulent sur le réseau. De part son importance, sa structure est conséquente: une antenne entre 5 et 7 mètres de diamètre, plusieurs baies remplies d'appareils et un prix unitaire d'environ 1 million d'euros. C'est aussi lui qui gère tous les accès à la bande passante.

Les stations *VSAT* permettent de connecter un ensemble de ressources au réseau. Dans la mesure où tout est géré par le hub, les points distants ne prennent aucune décision sur le réseau ce qui a permis de réaliser des matériels relativement petits et surtout peu coûteux. Dans la plupart des cas, une antenne d'environ 1 mètre permet d'assurer un débit de plusieurs centaines de Kb/s. Une station *VSAT* n'est donc pas un investissement important et l'implantation d'un nouveau point dans le réseau ne demande quasiment aucune modification du réseau existant. Ainsi une nouvelle station peut être implantée en quelques heures et ne nécessite pas de gros moyens. (Il suffit d'un technicien spécialisé).

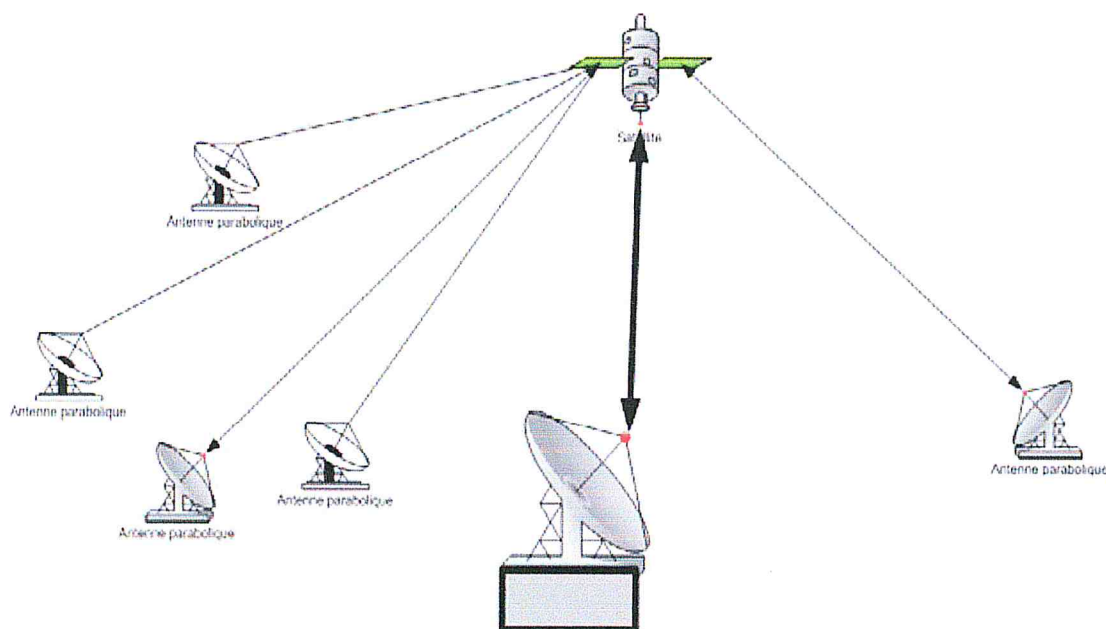


Figure I.2 : Illustration d'un réseau VSAT

3.2. Architectures des réseaux VSAT

Cette technologie utilise trois types d'architectures permettant un réseau point à point ou point à multipoints:

- ✓ Architecture en étoile où l'organisation du réseau repose autour du Hub (antenne principale).
- ✓ Architecture maillée où l'organisation ne repose pas sur le Hub mais offre la possibilité aux différents sites de communiquer entre eux (Interconnexion de deux ou plusieurs sites).
- ✓ Architecture mixte est une combinaison de l'architecture en étoile et de l'architecture maillée.

3.2.1. Architecture maillée

Dans cette architecture les VSATs sont reliés par le biais du canal satellite. Malheureusement dans le contexte de satellites géostationnaires (donc une distance de 35786 Km), les réseaux VSATs maillés subissent une atténuation de puissance de 200 dB sur la voie montante et sur la voie descendante.

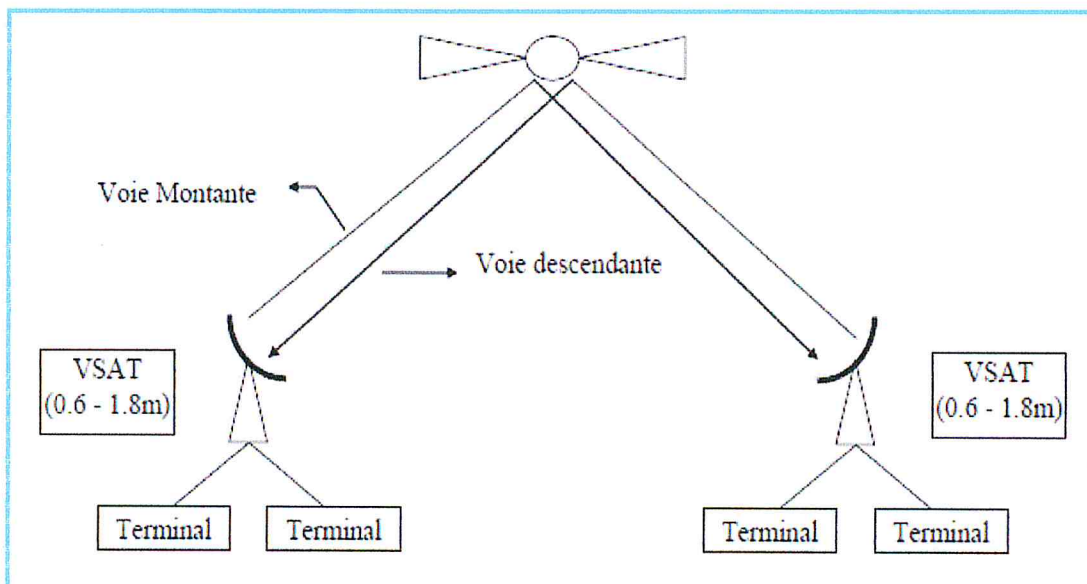


Figure 1.3 : Architecture d'un réseau maillé

3.2.2. Architecture en étoile

Pour résoudre le problème précédemment cité, un *HUB* a été introduit. Celui-ci a un diamètre supérieur à celui des *VSATs* (4-11m). L'architecture est dite 'en étoile'. L'ensemble de la voie montante et la voie descendante allant du VSAT émetteur vers le HUB est appelé 'inbound link' tandis que l'ensemble de la voie montante et la voie descendante allant du HUB vers le VSAT récepteur est appelée 'outbound link'.

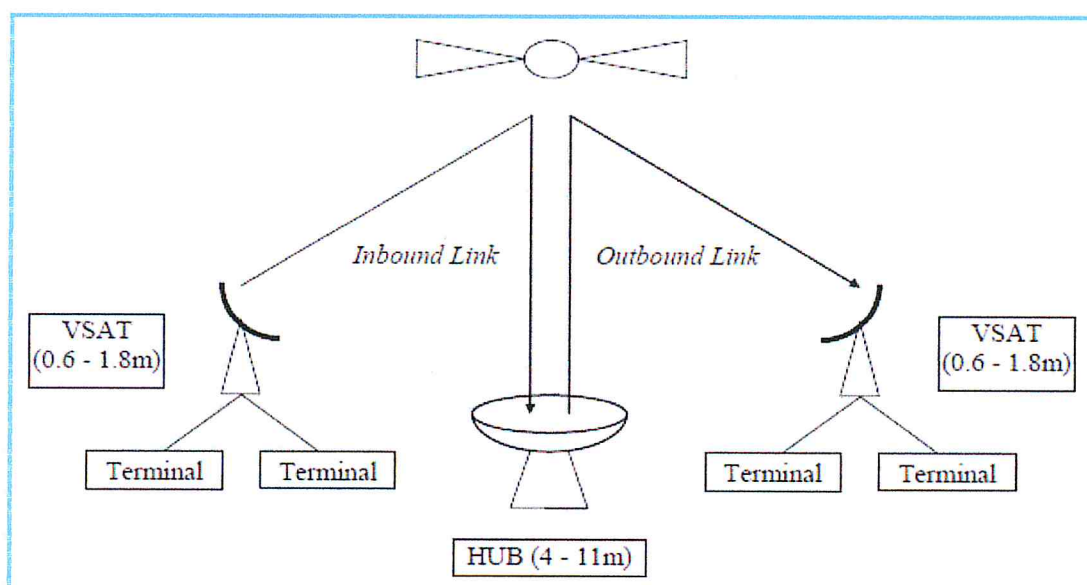


Figure 1.4 : Architecture d'un réseau en étoile

Ces réseaux peuvent être soit unidirectionnels soit bidirectionnels. Dans le cas d'un réseau unidirectionnel, les *VSATs* sont considérés comme des terminaux récepteurs seulement.

Cette architecture est utilisée pour des services de diffusion à partir d'un site central (le HUB) vers des sites distants (*VSATs*). Dans le cas d'un réseau bidirectionnel, les *VSATs* sont à la fois des émetteurs et des récepteurs.

3.2.3. Architecture mixte

Combine les deux types d'architectures en étoile et maillée. Les réseaux de télécommunication opérationnelle de la navigation aérienne sont basés sur ce type de configuration.

3.3. Gestion de la bande passante

Dans le cas des liaisons par satellites, la gestion de la bande passante est un élément très important car ce média est encore relativement cher. Si l'on loue un segment de 2 Mhz et que l'on se rend compte qu'en moyenne on ne consomme que 1 Mhz et bien on gaspille de la bande et par conséquent, on perd de l'argent. Certains types de liaisons comme les liaisons point à point sont des systèmes où l'on ne peut gérer la bande correctement. Mais ce n'est pas le cas du VSAT. Comme seul le point central gère l'accès au segment satellite, il est capable d'optimiser la gestion de la bande par un système de double multiplexage temporel et fréquentiel.

Voici un schéma qui représente un segment spatial divisé en différents canaux. La taille des canaux est fixée selon les débits qui sont désirés sur les stations VSAT:

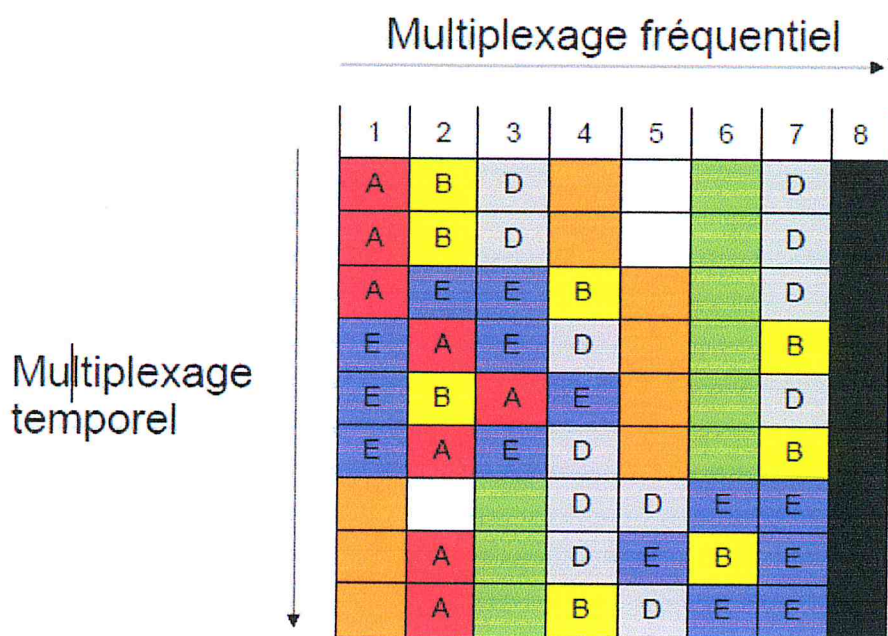


Figure I.3 : Illustration de la division d'un segment spatial en canaux

Dans le schéma ci-dessus, il y a huit canaux. Le dernier canal est un canal de signalisation qui permet aux stations de demander un canal au hub pour envoyer des données et qui permet au hub de dire aux stations sur quel canal écouter pour recevoir des données.

La gestion de la bande est propriétaire à chaque constructeur et chacun se garde bien de présenter sa méthode pour l'optimisation de la bande.

Le principe global est le suivant: lorsqu'une station veut avoir accès à une ressource, par exemple une requête sur une base de données se trouvent sur le site central, elle envoie une requête au hub pour que celui lui donne un canal pour envoyer les données. Et lorsque la base renvoie la réponse, le hub envoie un message à la station pour lui dire de prendre tel canal pour recevoir les données. Selon le débit défini, un point peut avoir accès à plusieurs canaux en même temps (multiplexage fréquentiel).

Comme on peut voir sur le schéma, un canal peut être partagé par plusieurs stations (multiplexage temporel). Comme l'adresse de destination figure dans le paquet, chaque station sait si les données qu'elle reçoit lui sont destinées ou pas (comme avec un hub sur un réseau Ethernet).

Ce système permet ainsi une forte optimisation de la bande passante ce qui réduit le coût du segment spatial à louer. [Web 1]

3.4. Les solutions VSAT [Web 4]

Le choix d'équipements satellite *VSAT* est souvent perçu de manière ésotérique par les architectes de réseaux. En effet, la dérégulation des télécommunications a fortement contribué à l'émergence d'une multitude de nouvelles techniques et de solutions dans ce domaine.

3.4.1. DVB /RCS

La norme DVB-RCS (Digital Video Broadcast - Return Channel System), soutenue par *ETSI (l'European Telecommunications Institute)* est une norme d'accès à Internet par satellite autorisant des débits allant jusqu'à 8 Mbits en flux descendant et jusqu'à 2 Mbps en flux montant. Elle a été développée en 1999 et décrit comment le trafic bidirectionnel des données devrait être transmis via satellite. Suivant le DVB-RCS, le terminal satellite du client peut recevoir une transmission standard de DVB (utilisé pour la diffusion de signaux vidéo ex : télévision via satellite) envoyée par la station satellite maîtresse (HUB). La transmission depuis l'installation client vers cette même station peut également être envoyée via la même antenne. La norme DVB-RCS définit une voie de retour au format MF-TDMA (Multi Frequency Time Division Multiplexing Access) qui permet de partager la capacité montante.

La norme spécifie que les liaisons montantes et descendantes doivent utiliser des fréquences différentes mais est indépendante des bandes de fréquences utilisées (Ku, Ka, L, S...).

Le DVB-RCS permet de transporter le protocole IP mais prend en compte également de nombreux protocoles de routage (RIP, IGMP) et de transport (RTP, UDP, TCPS).

3.4.2. SCPC

Le SCPC (Single Channel Per Carrier) est un système de transmission par satellite où chaque signal de télécommunication est modulé individuellement sur sa propre porteuse. La technologie SCPC est utilisée pour la transmission de données, pour la distribution simultanée du même message à plusieurs récepteurs et pour les communications audiovisuelles en mode full-duplex, c'est-à-dire de façon bidirectionnelle et simultanément en envoi et réception.

Dans un système SCPC, les signaux sont transmis en continu au satellite sur une unique porteuse. Le signal satellitaire est reçu dans un seul emplacement, pour le système point à point, ou dans plusieurs emplacements, pour le système point à multipoint, en réalisant ainsi une connectivité multi-site sans station centrale, c'est-à-dire complètement indépendante des stations terriennes.

3.5. Les applications

VSAT est un système qui est prévu pour mettre en place des réseaux de données. Mais depuis son apparition dans les années 80, des améliorations ont été apportées au système et les constructeurs ont réussi à augmenter considérablement le nombre d'applications possible avec un réseau de ce type.

Les terminaux VSAT possède des Slots permettant d'accueillir des cartes de différentes natures:

- ✓ Cartes réseaux : X25, FR, ATM, Ethernet, ...
- ✓ Cartes multimédia: Visioconférence, Streaming vidéo
- ✓ Cartes de communication: lignes analogiques, lignes numériques, ports séries

Grâce à toutes ces cartes, un réseau VSAT n'est plus seulement un réseau de données, mais il peut devenir un réseau téléphonique, un réseau de diffusion vidéo. Ces différentes technologies peuvent fonctionner en même temps ce qui accroît encore la modularité du système.

Voici un exemple possible de topologie VSAT utilisant différentes fonctionnalités fournies par le système:

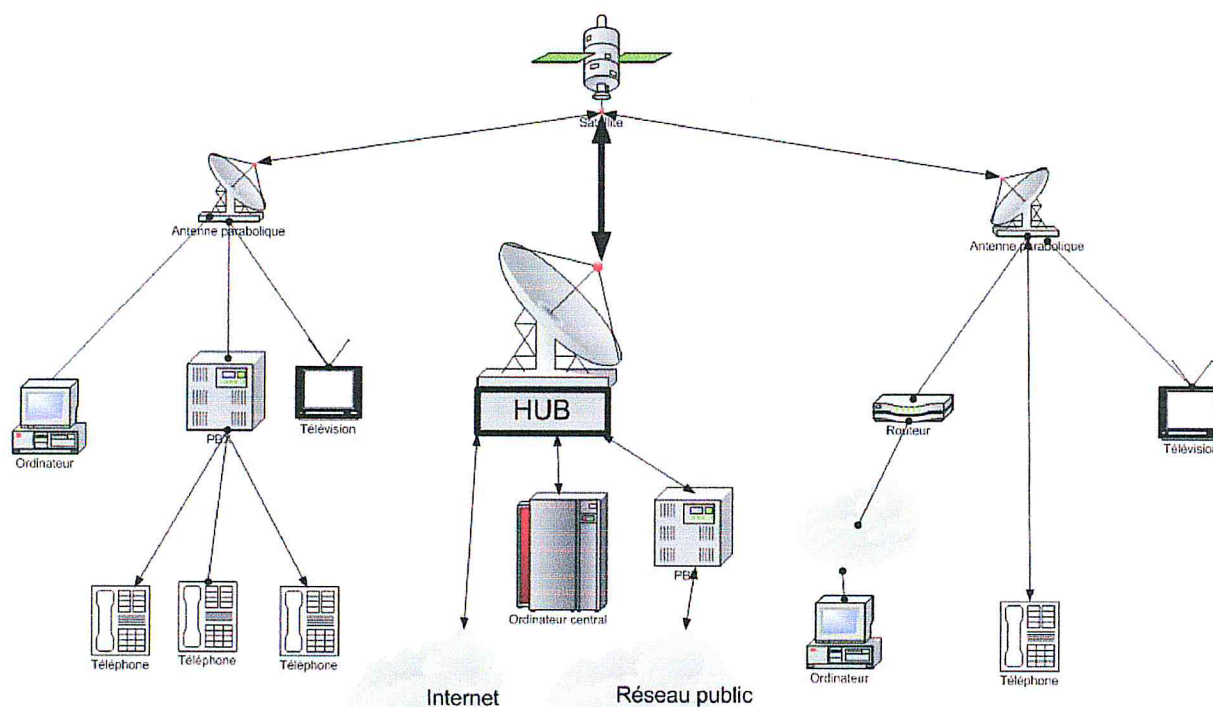


Figure I.3 : Les applications d'un réseau VSAT

4. Pourquoi choisir cette technologie

4.1. Quand penser à VSAT

La technologie VSAT permet de mettre en place des réseaux multi technologies à très grande échelle. Mais dans la mesure où cette technologie a un prix relativement élevée, elle est réservée aux grandes entreprises.

Lorsqu'une entreprise veut déployer un réseau, les premières questions qu'elle doit se poser sont : combien de points à connecter et où se situe chacun des points. Dans un pays comme la France, le réseau filaire est très développé, par conséquent, le prix d'une liaison loué est relativement abordable. Mais dans des pays qui ont une très grande superficie ou dans lesquels le réseau filaire est peu développé, un système comme le VSAT peut être une solution judicieuse car il n'y a la position géographique n'a plus guère d'importance, il faut juste voir le ciel.

Le nombre de points à connecter et lui aussi un facteur déterminant dans le choix de la technologie à utiliser. Avec un réseau de type filaire, il doit y avoir une LS par point vers le

La couverture d'un satellite géostationnaire à quelques exceptions près est fixe. Ceci veut dire que lorsqu'on a choisi un satellite, si une zone où un point doit être connecté prochainement n'est pas couverte, elle ne le sera jamais avec ce satellite. Alors que les réseaux filaires évoluent régulièrement ce qui laisse possible l'expansion d'un réseau dans des zones qui actuellement ne sont pas desservies.

Le fait que toutes les communications passent par le hub veut dire que si le hub tombe en panne tout le réseau est paralysé et plus aucune communication ne peut se faire.

5. Conclusion

5.1. L'importance de l'étude préliminaire

Comme il l'a été démontré précédemment, la technologie VSAT permet de mettre en place différents réseaux: de données, téléphoniques, vidéos. Comme ces réseaux peuvent fonctionner en même temps, il faut bien prévoir tous les équipements nécessaires à chacun pour permettre l'utilisation des différents réseaux.

L'investissement étant relativement important au départ, l'étude préliminaire ne doit oublier aucun paramètre afin de rentabiliser au maximum le système une fois qu'il sera en production.

5.2. Ou en est le VSAT

La technologie VSAT est apparue il y a une vingtaine d'années.

Au fur et à mesure des années, le système a été amélioré et sa démocratisation a permis de faire baisser les prix des matériels.

Aujourd'hui, certains opérateurs et fournisseur d'accès on fait l'acquisition de hubs et louent des accès pour que des entreprises qui n'ont pas les moyens de posséder leur propre hub. Ceci permet à des petites entreprises d'interconnecter plusieurs points pour un coût équivalent à un système filaire. Certains fournisseurs d'accès proposent des accès Internet pour les particuliers. Les débits et les tarifs sont équivalent à des systèmes filaires comme les connexions câble et XDSL.

Chapitre II : Généralités sur la gestion de réseau

1. Introduction

Administrer, c'est vouloir tirer le meilleur profit de la structure que l'on gère. Cependant, ce système est dual, car la conception d'une gestion dépend étroitement de la structure administrée. Inversement, le comportement futur de cette structure dépendra fortement de sa gestion. Les services et les équipements de télécommunication sont de plus en plus complexes et nombreux, cette évolution de la technologie met en évidence la nécessité de disposer d'architectures pouvant gérer ces services et contrôler ces ressources dans des environnements hétérogènes.

De façon générale, une administration de réseaux a pour objectif d'englober un ensemble de techniques de gestion mises en œuvre pour:

- ✓ Offrir aux utilisateurs une certaine qualité de service;
- ✓ Permettre l'évolution du système en incluant de nouvelles fonctionnalités;
- ✓ Rendre opérationnel un système.

2. Les disciplines de l'administration d'un système

L'administration de réseaux ne peut pas être isolée de son environnement. En effet il s'agit d'une partie intégrante d'un système plus général constitué de trois parties fondamentales qui sont:

- ✓ Les utilisateurs ou consommateurs de services;
- ✓ Les serveurs d'applications ou fournisseurs de services;
- ✓ La machine de transport reliant les utilisateurs aux fournisseurs.

Ces trois groupes déterminent les trois disciplines d'administration d'un système [SIM 97]:

2.1. L'administration des utilisateurs

Qui fournit l'ensemble des mécanismes pour:

- ✓ **L'accessibilité et la connexion aux applications** : en effet, l'utilisateur doit pouvoir se connecter aux différentes applications et, pour cela, il doit disposer d'un ensemble d'outils qui lui assurent la transparence des méthodes d'accès et de connexion aux différentes applications;

- ✓ **L'accès aux serveurs de noms** : afin de permettre la localisation des ressources et d'assurer à l'utilisateur l'existence et l'utilisation de ces ressources.
- ✓ **La confidentialité et la sécurité**: le système doit fournir l'ensemble des mécanismes qui permettent de garantir la confidentialité des informations de l'utilisateur, de sécuriser son environnement et de prévenir toute perte ou altération des échanges effectués par l'utilisateur;
- ✓ **La qualité de service** : qui est la perception du réseau ressentie par l'utilisateur. Son exigence majeure concerne la disponibilité et les performances du système ainsi que sa capacité à assurer un service convenable.

2.2. L'administration des serveurs

Qui recouvre l'ensemble des mécanismes à mettre en place pour:

- ✓ **La connexion et la distribution des applications** :

Cela afin de permettre l'interrelation des services;

- ✓ **La gestion et la distribution des données** :

Qui, comme pour les utilisateurs, doivent garantir la fiabilité de transmission des informations sur le réseau et offrir un certain nombre d'outils permettant le transfert de ces informations. C'est typiquement le rôle des outils de transfert de fichiers, qui permettent le partage des capacités de stockage entre plusieurs systèmes;

- ✓ **La gestion des applications** :

Correspond essentiellement au contrôle et la protection des accès à ces applications par la distribution de droits, ainsi qu'à la fourniture de mécanismes de contrôle d'utilisation de ressources concernant l'application.

2.3. L'administration de la machine de transport

Qui consiste à fournir des mécanismes sur:

- ✓ Des opérations de réseau, dont le rôle est de permettre l'intervention sur le fonctionnement du réseau et la modification si nécessaire;
- ✓ Les incidents réseaux par la mise en place de mécanismes de détection et de correction. Lorsqu'une alerte est déclenchée, des actions doivent être prises pour résoudre rapidement l'incident et réduire son influence sur la qualité de service;

3.2 L'interprétation des données :

Une fois que l'on a les informations de gestion, il est important de les interpréter correctement. Par exemple : 5000 paquets par seconde sur un port d'un routeur est peut-être normal ou peut-être le signe d'une surcharge et d'une dégradation de la qualité du service offert. Même si on connaît la charge exacte d'un routeur, cette information ne suffit pas pour prendre des décisions pertinentes de gestion.

Les fabricants d'équipements offrent quelquefois des logiciels qui savent interpréter les informations de gestion de leurs équipements de gestion. Toutefois, une interprétation automatique complète et correcte de l'état d'un réseau et de ses équipements est difficile à réaliser. L'intervention humaine est souvent requise pour interpréter les données ou pour placer des bornes acceptables.

3.3 Le contrôle des équipements

Quand des informations de gestion sont obtenues et comprises, il est quelquefois nécessaire d'agir. Il doit être possible de demander à un équipement, par exemple, de se réinitialiser, de couper ou d'activer des services...

Pour ce faire, un protocole de gestion doit transmettre un ordre (requête) à l'équipement approprié pour déclencher l'action. En raison de l'absence de sécurité par le passé, cette fonctionnalité a rarement été utilisée.

4. Principes de la gestion de réseau

Les architectures de gestion de réseau sont classées dans la catégorie des systèmes client/serveur [Web 6]. On peut donc identifier trois composants:

✓ La partie client :

Appelée manager, est l'entité qui va superviser les opérations de gestion. Elle sert d'interface entre l'utilisateur et le serveur. Il peut y avoir, le cas échéant, plusieurs managers.

✓ La partie serveur

Appelée agent représente un ou plusieurs équipements. Un agent sert d'interface entre l'équipement et le manager. Il peut également servir d'interface entre plusieurs autres agents et le manager. L'agent permet de fournir une vue abstraite de l'équipement et masque ainsi les différences qui existent entre les équipements. Il y a bien entendu plusieurs agents dans un réseau.

✓ Le protocole

Utilisé par le manager et l'agent pour communiquer (SNMP, CMIS, CMIP, CMOT, ...) Il est mis en oeuvre par le biais d'un réseau. Ce protocole permet au manager d'interroger l'agent (dans le cas du monitoring) ou de lui demander de modifier l'état de l'équipement (dans le cas de la configuration). Le protocole est également utilisé par l'agent afin d'avertir le manager de l'occurrence d'événements.

Nous allons étudier dans ce qui suit le protocole le plus utilisé par les logiciels de supervision à savoir le protocole SNMP.

5. Le protocole SNMP

L'activité de recherche et de standardisation au niveau de SNMP est menée sous l'égide de l'IETF (Internet Engineering Task Force). La première version de SNMP a vu le jour en 1989. La seconde version, appelée SNMP v2 a été standardisée en 1992 [RFC1905]. Cette version apporte essentiellement des changements au niveau de la sécurité et de la confidentialité des opérations de gestion. Suite à des problèmes apparus lors de la mise en opération de la seconde version du protocole, ces aspects ont été classés obsolètes en 1996 lors d'un des meetings tri annuel de l'IETF. Ce dernier rebondissement a eu pour effet, d'une part de renvoyer SNMP à son état de 1989 et d'autre part, de déclencher les recherches sur SNMP v3. Les travaux se poursuivent actuellement et petit à petit les différents éléments de l'architecture SNMP v3 sont proposés à la communauté scientifique [RFC2274]. SNMP v3 (dans les aspects qui sont actuellement définis) est en passe de devenir un standard car il est accepté par le working group SNMP v3 en tant que *proposed standard*. A l'heure actuelle, SNMP v1 reste la version la plus utilisée. De plus, les résultats obtenus avec SNMP v1 sont immédiatement applicables à SNMP-v2. [Web 6]

5.1. Composants du protocole SNMP

Le modèle SNMP définit deux entités de base qui forment un modèle client serveur. L'élément serveur s'appelle un agent SNMP et est installé dans l'ensemble des équipements informatiques à superviser. L'élément client s'appelle un manager SNMP ou station de supervision. L'agent SNMP écoute les requêtes du manager SNMP sur le port UDP 161 et le manager SNMP écoute les alarmes ou traps de l'agent SNMP sur le port UDP 162.

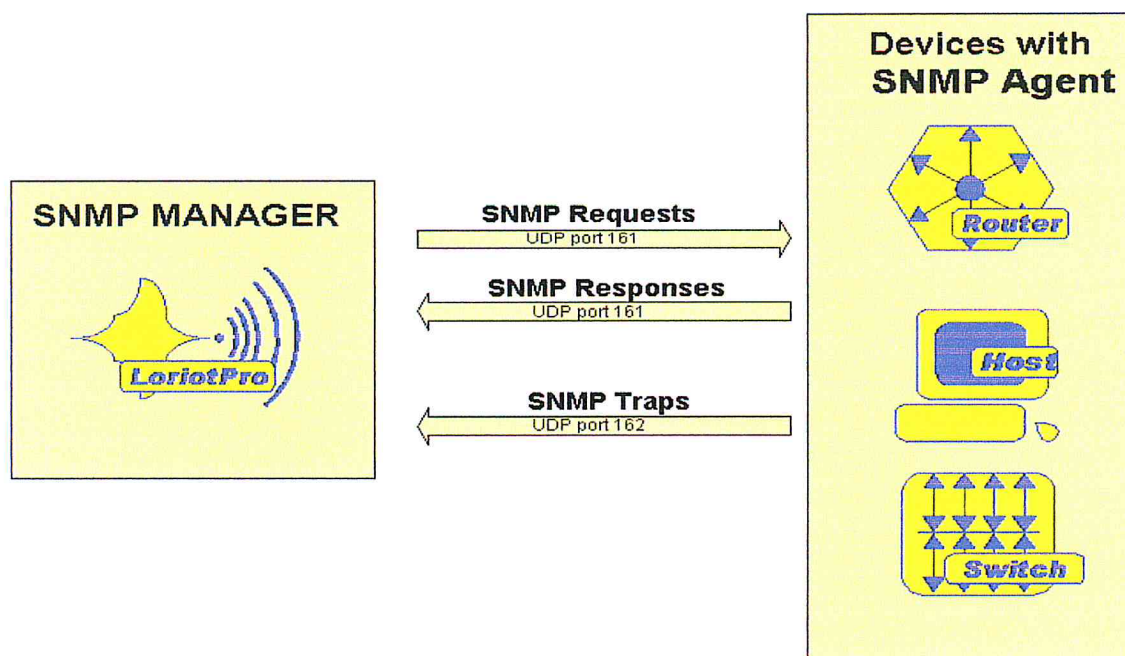


Figure II.2 : Architecture du protocole SNMP

5.1.1. Le manager SNMP

Le manager SNMP est généralement installé sur une station de travail performante connectée au réseau d'entreprise. Son rôle est d'acquies à l'aide de requêtes SNMP des informations propres aux différents agents composant le réseau. Les informations recueillies sont ensuite traitées et présentées suivant la qualité de l'interface utilisateur sous forme de tableau, courbe, histogramme, pour une interprétation plus aisée. Le manager SNMP est plus communément appelé *Station de management* ou de *supervision*. La station de supervision reçoit les *traps* (messages d'information ou d'alarme) émis par l'*agent* à travers un socket UDP ouverte sur le port 162.

Le manager SNMP se compose des éléments suivants :

- **L'interface utilisateur** : Généralement, elle utilise une version améliorée du GUI (Graphical User Interface) du système d'exploitation.
- **Les applications de gestion** : Elles aident à analyser et représenter les informations de gestion obtenues des agents. En général ce sont des tableaux, courbes, histogrammes ... La station de supervision ouvre un socket UDP en écoute sur le port 162 pour permettre la gestion des *Traps* émis par les agents.

- **Les bases de données** : La station de supervision a la capacité d'enregistrer toutes les informations et statistiques du réseau dans une base de données partagée ou des fichiers de données spécifiques.
- **Le protocole de transport** : C'est l'ensemble des protocoles qui va assurer la transmission de datagrammes entre les équipements du réseau. Les protocoles les plus usités actuellement sont IP et UDP. On dit que la station possède une pile de protocoles TCP/IP.
- **Le moteur SNMP** : C'est le noyau du logiciel de supervision ; il fournit l'accès au protocole de communication avec les agents, ce qui permet la collecte des informations. Actuellement le protocole SNMP existe en plusieurs versions qui vont de la version 1 à la version 3.
- **Les profils de management des agents** : C'est un ensemble de règles qui définit l'accès aux différents agents du réseau, l'ensemble de ces règles permet d'obtenir la Map du réseau (représentation graphique de la topologie du réseau).

5.1.2. L'agent SNMP

L'agent SNMP est situé dans l'équipement à contrôler. La plupart des équipements du réseau sont équipés de la portion de logiciel qui les transforme en agent SNMP. De base, tous les équipements d'interconnexion tels que les routeurs sont des agents. La plupart des serveurs possèdent des capacités SNMP. Les Hubs, suivant leur niveau de version matériel et logiciel, sont eux aussi administrables. L'agent répond aux requêtes SNMP du manager à travers un socket UDP ouverte sur le port 161.

Un agent SNMP se compose des éléments suivants :

- **Le protocole de transport** : C'est l'ensemble des protocoles qui vont assurer la transmission de datagrammes entre les équipements du réseau. Les protocoles les plus usités actuellement sont IP et UDP. On dit que l'équipement possède une pile de protocoles TCP/IP.
- **Le moteur SNMP**: C'est le processus logiciel qui implémente le protocole SNMP dans ses différentes versions.
- **Le profil de management**: C'est un ensemble de règles qui définit l'accès aux variables de la MIB de l'agent et indique les paramètres des stations de supervision autorisées.

Chaque agent SNMP a un ensemble d'objets associés pouvant être interrogé par le manager SNMP. Un objet représente un élément physique ou logique de l'équipement ou un groupe d'éléments (table). Il doit pouvoir répondre à un ensemble prédéfini d'objets pour prétendre être compatible.

Pour simplifier la lecture des objets contenus dans l'agent, une structure d'arbre standard est utilisée pour regrouper l'ensemble des objets par blocs ou branches. Cette structure contient des objets de différents types permettant de créer les nœuds de l'arbre et des feuilles qui contiennent en plus du nom des objets, des valeurs que le logiciel de supervision pourra collecter.

L'accès par un manager SNMP à un objet d'un agent SNMP doit toujours préciser un numéro d'index. Un index à valeur 0 permet d'accéder à un objet non indexé. Par contre tous les objets situés dans des tables d'objets SNMP possèdent un index distinct. Lors de l'utilisation d'équipements ayant plusieurs fois le même type d'élément interne, plusieurs cartes réseaux par exemple, il faut alors spécifier le numéro de la carte pour obtenir des informations relatives à celles-ci. Cette sélection est faite par le numéro d'index.

5.1.3. La MIB (Management Information Base)

Les *MIB* sont les fichiers descriptifs des objets utilisés par le protocole SNMP et implémenté dans les agents SNMP.

Le terme MIB, pour base de données des informations de gestion réseau, est utilisé car la structure de ces fichiers est très similaire à une structure de base de données.

Un fichier de MIB est un fichier au format texte, contenant des définitions. Le document RFC 1155 définit les règles d'écriture en SMI V1 (Structure of Management Information) d'un fichier de MIB et le document RFC 1213 contient la définition des objets devant être implémentés dans un agent. Les objets de la MIB sont organisés hiérarchiquement dans une structure d'arbre et selon un ordre défini dans des documents de standardisation appelés des RFC (Request For Comments).

On peut classer les MIBs en deux principaux groupes :

⌘ MIB Standards :

Editées par des organismes de normalisation internationaux, ces principales informations sont contenues dans les RFC 1157/1155/1213.

Deux MIB publics ont été normalisées: MIB I et MIB II (dite 1 et 2) :

MIB I : Contient les variables globales telles que le nom de l'équipement, sa position géographique, le nombre de paquets transmis, etc...

MIB II : Contient les informations de la MIB I et toutes les variables relatives à SNMP.

Les objets des MIBs standards sont attachés à la branche MIB-2 de l'arbre et ils sont définis et mis à jours par les groupes de travail correspondant aux différentes technologies.

⌘ MIB propriétaires :

Ces objets sont attachés à la branche *private* de l'arbre et ils sont définis par les constructeurs d'équipements. Chaque constructeur doit faire la demande d'un numéro d'entreprise lui permettant de s'insérer dans l'arbre global des MIBs.

Les constructeurs ayant des spécificités et voulant élargir la capacité de management de leurs équipements ont développé leurs propres MIBs.

Toutes les MIBs constructeur sont accessibles par le chemin :

iso(1).org(3).dod(6).internet(1).private(4).entreprises.xxx

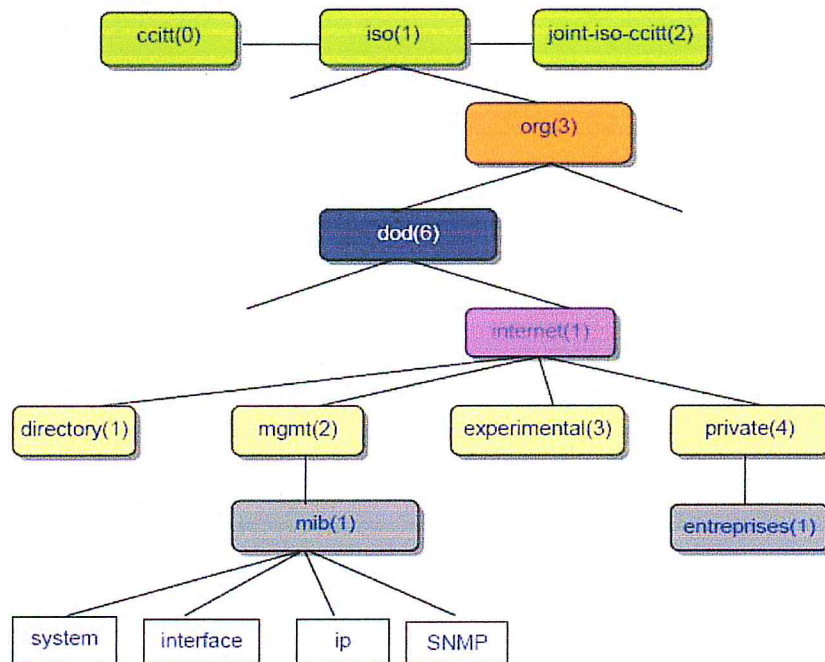


Figure II.2 : Structure d'indexation des données dans la MIB SNMP

Les objets et instances sont nommés par un identificateur que l'on appelle *object identifier*. Il s'agit d'une suite de nombres séparés par des points. La dot notation fournit un chemin dans l'arbre à partir de la racine. La racine est identifiée par le nombre '1'. Le nombre suivant, c'est-à-dire le second nombre dans la dot notation, identifie le numéro du fils de la racine. On

procède récursivement jusqu'à la fin de *l'object identifier*. A titre d'exemple, l'objet *snmpInPkts* possède comme *object identifier* 1.3.6.1.2.1.1.1.1 Les cinq premiers identificateurs sont fixes car ils correspondent à la racine de toute l'arborescence de management:

iso(1).org(3).dod(6).internet(1).mgmt(2)

5.2. Fonctionnement du protocole SNMP

Le protocole SNMP permet à la console de management de communiquer avec les agents. Il utilise des protocoles de transport tel que IP et IPX. Le principal rôle du protocole SNMP va être de permettre la collecte des objets contenus dans les agents.

On peut découper l'étude du protocole SNMP en trois parties :

- Le protocole lui-même qui va définir les méthodes d'échanges d'informations, le format des paquets de données, les différents types de requêtes, la sécurisation des échanges.
- La structure de l'information contenue dans les agents et les méthodes de stockage de cette information. Le terme anglais pour cette partie est SMI pour *Structure of Management Information* qui fait l'objet de plusieurs publications.
- Enfin la structure de l'ensemble des objets maintenus par un agent et la définition de cette structure appelée MIB pour Management Information Base.

5.2.1. Les requêtes SNMP

Le mécanisme de base du protocole SNMP est constitué d'échanges de type requête/réponse appelé PDU (*Protocol Data Unit*). En fonction de la version du protocole SNMP utilisé, différentes commandes sont possibles. Les requêtes SNMP vont contenir une liste d'OID (object identifier) à collecter sur l'agent SNMP.

Le manager SNMP envoie quatre types de requêtes vers l'agent SNMP :

- ***Get Request*** : Le manager interroge un agent sur les valeurs d'un ou de plusieurs objets d'une MIB ;
- ***Get Next Request*** : Le manager interroge un agent pour obtenir la valeur de l'objet suivant dans l'arbre des objets de l'agent. Cette interrogation permet de balayer des objets indexés de type tableau ;

- **Get Bulk Request** : Introduite avec la version 2 du protocole SNMP, cette requête permet de mixer la commande *GET* et *GETNEXT* pour obtenir des blocs entiers de réponses de la part de l'agent ;
- **Set Request** : Le manager positionne ou modifie la valeur d'un objet dans l'agent.

L'agent envoie trois types de réponse vers le manager :

- **Get Response** : L'agent répond aux interrogations du manager.
- **Trap** : L'équipement génère un envoi vers son manager pour signaler un événement, un changement d'état ou un défaut. L'agent n'attend pas d'acquittement de la part du manager.
- **Notification** : Introduite avec la version 2 du protocole SNMP. L'équipement génère un envoi vers son manager pour signaler un événement, un changement d'état ou un défaut. L'agent n'attend pas d'acquittement de la part du manager.
- **Inform** : Introduite avec la version 2 du protocole SNMP. L'équipement génère un envoi vers son manager pour signaler un événement, un changement d'état ou un défaut. L'agent attend un d'acquittement de la part du manager et il y aura une retransmission en cas de non réponse.

6. Les outils de supervision réseau

On ne peut pas réellement parler de supervision, mais quand même de surveillance en utilisant des petits outils simples qui, une fois combinés à l'aide de langages de scripts permettent de réaliser des choses très intéressantes. D'autres part, ces petits outils sont la base des plus grosses solutions de supervision, il semble donc intéressant d'appréhender leur fonctionnement [Web 7].

6.1. Outils SNMP

Il existe bien évidemment des programmes simples destinés à utiliser le protocole SNMP. Leurs noms indiquent quelle requête ils peuvent envoyer, par exemple : *snmpget*, *snmpgetnext*, *snmpinform*, *snmpset* etc...

Chacune de ses commandes permet de réaliser la requête en utilisant une des trois versions du protocole SNMP. Elles sont énormément utilisées dans le cadre de scripts qui stockent les valeurs obtenues afin de réaliser des statistiques ou des alertes.

6.2. MRTG

MRTG est un outil réalisé en Perl et en C dans le but de surveiller la charge des liens réseaux. Il génère des pages html contenant des images au format PNG qui représentent graphiquement l'état en temps réel de la ressource surveillée. Le principe est simple : un script Perl recherche les données via le protocole SNMP et envoie celles-ci à un programme C qui va les stocker et générer les graphiques

A la base l'auteur avait dans le but de surveiller le trafic passant par des routeurs, mais MRTG se basant sur SNMP, les possibilités se sont étendues à toute variable. Encore mieux, on peut aussi créer un script qui surveillera n'importe quel type de donnée non disponible dans SNMP. On possède ainsi un système de surveillance déjà conséquent qui permet sur une même page de surveiller un réseau et de garder les traces des anciennes données.

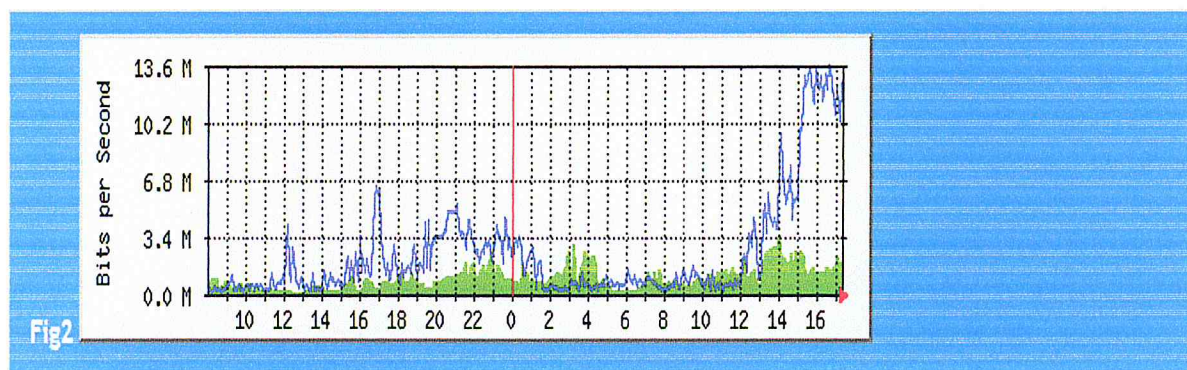


Figure II.3 : Génération de graphiques dans MRTG

6.3. RRDTOOL

Du même auteur que MRTG, RRDTOOL est un peu considéré comme une évolution de ce dernier. La gestion de stockage des données a surtout été entièrement revue pour améliorer les performances, pour cela l'auteur a utilisé une technique nommée "Round Robin", d'où le nom "RRDTOOL" (Round Robin Database TOOL).

Les graphiques se sont également améliorés offrant plus de possibilités, et l'interfaçage entre la récupération de donnée et la génération des images est devenu plus modulaire

permettant de réaliser des scripts dans beaucoup de langages différents. RRDTOOL est maintenant un outil incontournable pour générer des graphiques, il est donc utilisé dans quasiment tous les logiciels de supervision open-source.

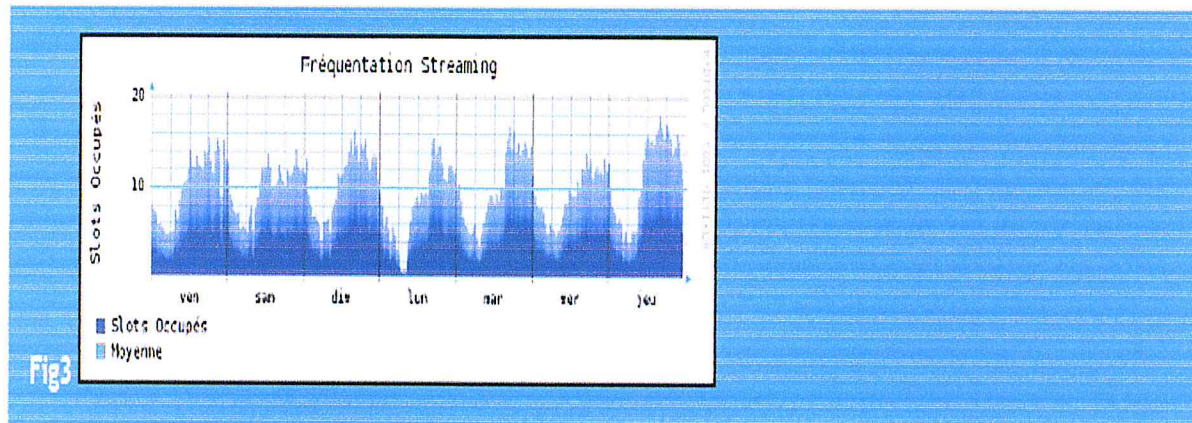


Figure II.4 : Génération de graphiques dans RRDTOOL

Il existe bien entendu des centaines d'autres programmes très utiles pour surveiller un parc informatique. On peut par exemple citer le très utile *TcpDump* permettant de capturer les paquets circulant sur le réseau qui pourra être pour détecter avec un peu de *scripting* des utilisations frauduleuses ou excessives du réseau.

Iperf en est un autre assez utilisé pour réaliser une sorte d'audit du réseau. Bien sur, les possibilités sont très vastes puisque on peut également réaliser nos propres outils en ayant quelques connaissances sur les protocoles réseau et l'ouverture de socket dans un langage quelconque.

L'ensemble des programmes présentés jusqu'ici représentent des briques de base que vont utiliser des logiciels de supervision plus complet dans le but de créer une console d'administration permettant de gérer l'ensemble des machines d'un réseau. Les administrateurs ayant de plus en plus de machines à gérer, les applications de supervision doivent permettre le passage à l'échelle de manière aisée : Il doit être possible de passer d'une vue globale de centaines de machines pour savoir si il y a une erreur quelque part et pouvoir "zoomer" sur la machine en question et obtenir toutes les informations précises sur celle-ci afin de connaître l'origine du problème.

7. Conclusion

L'objectif des administrateurs d'un système d'information est de pouvoir connaître en permanence l'état du réseau, et d'être averti en temps réel des différents incidents pour réduire au maximum les délais d'intervention et de coupure du service.

A présent, de nouveaux besoins se font sentir. Le premier besoin est de rompre la contrainte de proximité de l'équipement à gérer. Avec les protocoles de gestion de réseau actuels, le réseau est utilisé afin de permettre la gestion des équipements le composant. On se trouve donc dans le cas d'une gestion du réseau par le réseau. Le second besoin concerne la dépendance humaine. La complexité des réseaux s'accroissant, la tâche de gestion du réseau exige une telle masse d'informations à traiter qu'elle dépasse la capacité de l'être humain. En effet, il faut remarquer que chaque information individuelle sur un équipement ne peut être utilisée telle quelle, elle n'est significative que dans le contexte du réseau global.

Les considérations énoncées ci-dessus permettent d'identifier les besoins de la gestion des réseaux:

- ✓ S'adapter à toutes les tailles de réseaux.
- ✓ Gérer des équipements hétérogènes et possiblement complexes.
- ✓ Permettre les opérations de gestion à distance.
- ✓ Assister l'utilisateur en:
 - Automatisant certaines tâches.
 - Aidant l'utilisateur dans les tâches qui ne sont pas automatisées.
 - Fournissant une vue adaptée du réseau selon les opérations à réaliser.

Chapitre III: Méthode d'analyse et de conception

1. Introduction

Depuis quelques années, la modélisation objet avec le langage UML est devenue incontournable sur la plupart des projets informatiques. Alors pourquoi ne pas appliquer aux projets web ce qui marche pour les projets « classiques » ? Contrairement à une idée répandue, les applications web sont justement, de par leur complexité croissante, des candidates idéales à la modélisation graphique et à l'application d'un processus de développement formalisé.

Dans ce chapitre introductif, nous présenterons brièvement le langage de modélisation UML et le processus de développement que nous avons adopté pour la modélisation de notre application, à savoir le processus *2TUP* (*2Track Unified Process*).

2. Le langage UML

UML ou *Unified Modeling Language*, est un langage de modélisation qui est né au milieu des années 90 de la fusion de trois méthodes objets: *OMT* (*Object Modeling Technique*), *BOOCH* et *OOSE* (*Object Oriented Software Engineering*).

L'idée de cette fusion est partie du constat qu'à l'époque il existait plusieurs méthodes objets liées par un consensus autour d'idées communes: objets, classes, sous-systèmes etc.

C'est à partir de 1997 que *l'OMG* (*Object Management Group*) qui standardise les technologies de l'objet, s'est attachée à la définition d'un langage commun unique, utilisable par toutes les méthodes objets dans toutes les phases du cycle de vie et compatible avec les techniques de réalisation du moment. D'où la naissance d'UML [Web 8]. UML offre des éléments pour décrire les différents aspects d'un système: les diagrammes. Les diagrammes utilisés dans l'ensemble de notre analyse sont:

- ✓ Le diagramme de contexte
- ✓ Le diagramme des cas d'utilisation
- ✓ Le diagramme de classes d'objets
- ✓ Le diagramme de séquence

3. Les processus de développement logiciel

3.1. Processus unifié

Un processus unifié est un processus construit sur UML (Unified Modeling Language). Les processus unifiés sont le résultat de l'unification, non pas des processus, mais plus exactement des meilleures pratiques du développement objet [Web 9].

Un processus unifié se distingue par les caractéristiques suivantes :

- ✓ **Itératif** : Le logiciel nécessite une compréhension progressive du problème à travers des raffinements successifs et développer une solution effective de façon incrémentale par des itérations multiples.
- ✓ **Piloté par les risques** : les causes majeures d'échec d'un projet logiciel doivent être écartées en priorité.
- ✓ **Centré sur l'architecture** : le choix de l'architecture logicielle est effectué lors des premières phases de développement du logiciel. La conception des composants du produit est basée sur ce choix.
- ✓ **Conduit par les cas d'utilisation** : le processus est orienté par les besoins utilisateurs présentés par des cas d'utilisation.

Le tableau suivant présente une synthèse des processus les plus en vogue dans la communauté objet et nouvelles technologies.

	Description	Points forts	Points faibles
RUP <i>Rational Unified Process</i>	<ul style="list-style-type: none"> - Promu par Rational. - Le RUP est à la fois une méthodologie et un outil prêt à l'emploi (documents types partagés dans un référentiel Web) - Cible des projets de plus de 10 personnes 	<ul style="list-style-type: none"> - Itératif - Spécifie le dialogue entre les différents intervenants du projet : les livrables, les plannings, les prototypes... - Propose des modèles de documents, et des canevas pour des projets types 	<ul style="list-style-type: none"> - Coûteux à personnaliser - Très axé processus, au détriment du développement : peu de place pour le code et la technologie
XP <i>eXtreme Programming</i>	<ul style="list-style-type: none"> - Ensemble de « Bests Practices » de développement (travail en équipes, transfert de compétences...) - Cible des projets de moins de 10 personnes 	<ul style="list-style-type: none"> - Itératif - Simple à mettre en œuvre - Fait une large place aux aspects techniques : prototypes, règles de développement, tests... 	<ul style="list-style-type: none"> - Ne couvre pas les phases en amont et en aval au développement : capture des besoins, support, maintenance, tests d'intégration... - Elude la phase d'analyse, si bien qu'on peut dépenser son énergie à faire et défaire
2TUP <i>Two Track Unified Process</i>	<ul style="list-style-type: none"> - S'articule autour de l'architecture - Propose un cycle de développement en Y - Cible des projets de toutes tailles 	<ul style="list-style-type: none"> - Itératif - Fait une large place à la technologie et à la gestion du risque - Définit les profils des intervenants, les livrables, les plannings, les prototypes 	<ul style="list-style-type: none"> - Plutôt superficiel sur les phases situées en amont et en aval du développement : capture des besoins, support, maintenance, gestion du changement... - Ne propose pas de documents types

Tableau III.1: Synthèse des méthodologies utilisées dans le cadre de développement Objet et nouvelles technologies [Web 9].

3.2. Le processus 2TUP

3.2.1. Définition [Web 10]

2TUP « 2 Track Unified Process », est une variante du processus unifié, donc qui répond aux caractéristiques que nous avons cité précédemment. Le processus 2TUP apporte une réponse aux contraintes de changement continu imposées aux systèmes d'information de l'entreprise. En ce sens, il renforce le contrôle sur les capacités d'évolution et de correction de tels systèmes.

« 2 Track » signifie littéralement que le processus suit deux chemins. Il s'agit des chemins « fonctionnels » et « d'architecture technique », qui correspondent aux deux axes de changement imposés au système informatique.

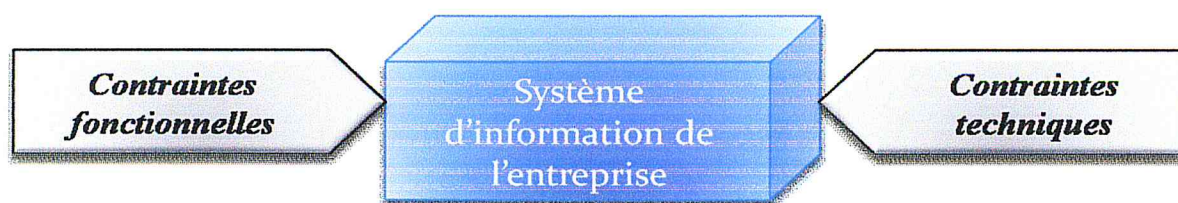


Figure III.1: Les contraintes soumises aux systèmes

Le processus 2TUP propose un cycle de développement en Y qui dissocie les aspects techniques des aspects fonctionnels et propose une étude parallèle des deux branches

Illustré sur la figure suivante, ce cycle s'articule autour de trois phases :

- ✕ Une branche technique
- ✕ Une branche fonctionnelle
- ✕ Et une branche de conception réalisation.

La figure suivante détaille les étapes de développement des trois branches du processus 2TUP.

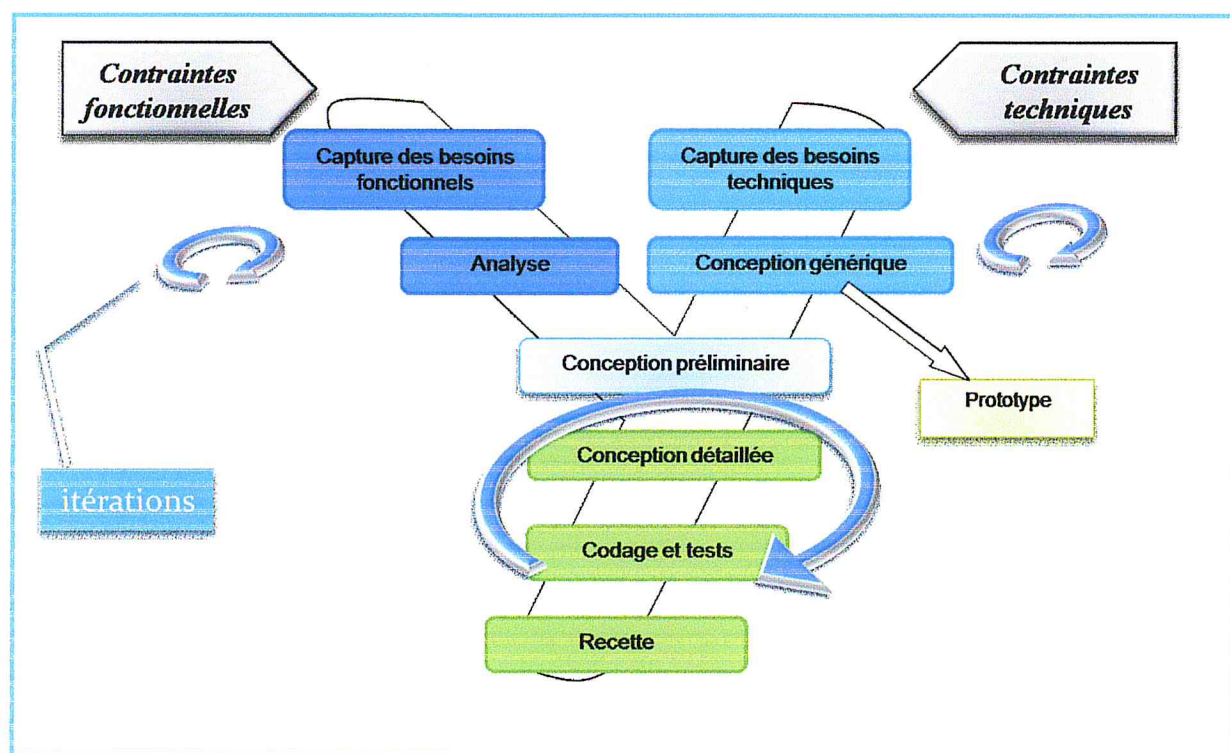


Figure III.2 : Le processus de développement en Y

3.2.2. Branche fonctionnelle [ROQ2003]

Les principales étapes de la branche fonctionnelle se présentent comme suit :

✓ L'étape **capture des besoins fonctionnels** produit le modèle des besoins focalisé sur le métier des utilisateurs. Elle qualifie, au plus tôt le risque de produire un système inadapté aux utilisateurs. Cette phase a pour objectif de définir :

- ✕ La frontière fonctionnelle entre le système considéré comme une boîte noire et son environnement, c'est le niveau contexte.
- ✕ Les activités attendues des différents utilisateurs par rapport au système toujours envisagé comme une boîte noire, c'est le niveau cas d'utilisation.

✓ L'étape d'**analyse** consiste à étudier précisément les spécifications fonctionnelles de manière à obtenir une idée de ce que va réaliser le système en terme de métier. Les résultats de l'analyse ne dépendent d'aucune technologie particulière.

3.2.3. Branche technique

Les principales étapes de la branche technique se présentent comme suit :

✓ L'étape **capture des besoins techniques** recense toutes les contraintes sur les choix de dimensionnement et la conception du système. Les outils et le matériel sélectionnés ainsi que la prise en compte des contraintes d'intégration avec l'existant (pré requis d'architecture technique).

Cette étape permet de définir le modèle d'analyse technique. Le rôle de ce dernier est d'établir les couches logicielles et y spécifie les activités techniques attendues.

✓ L'étape **conception générique** définit ensuite les composants nécessaires à la construction de l'architecture technique. Cette conception est complètement indépendante des aspects fonctionnels. Elle permet de générer le modèle de conception technique ou design pattern qui définit les Frameworks. Ces derniers, délivrant les services techniques, assurent la réponse aux exigences opérationnelles du système.

3.2.4. Branche conception/réalisation

Les principales étapes de cette branche se présentent comme suit :

✓ L'étape **conception préliminaire** est une étape délicate, car elle intègre le modèle d'analyse fonctionnelle dans l'architecture technique de manière à tracer la cartographie des composants du système à développer.

Cette étape permet de produire le modèle de conception système. Ce dernier organise le système en composants, délivrant les services techniques et fonctionnels. Ce modèle regroupe les informations des branches technique et fonctionnelle.

- ✓ L'étape **conception détaillée** permet d'étudier comment réaliser chaque composant.

Cette étape produit le modèle de conception des composants. Ce modèle fournit l'image prête à fabriquer du système complet.

- ✓ L'étape de **codage** ou s'effectue la production des composants et les testes des unités de code au fur et à mesure de leur réalisation.
- ✓ L'étape de **recette** consiste à valider les fonctionnalités du système développé

4. Conclusion

Dans le cadre d'un projet de développement informatique, il apparaît aujourd'hui une nouvelle dimension qui est celle de la « réutilisation » : réutilisation des spécifications, des services, des composants et du code.

Cette démarche permettra un gain considérable de temps et donc d'argent, l'amélioration de la qualité des développements et donc satisfaction des utilisateurs et une meilleure réactivité lors d'une demande de modification, donc "Agilité".

Afin d'atteindre ces objectifs notre choix s'est porté sur le processus de développement *2TUP* qui apporte une méthodologie permettant de disséquer un projet en plusieurs parties fortement réutilisables.

Chapitre IV: Etude préliminaire

1. Introduction

L'étude préliminaire est la toute première étape de notre processus de développement. Elle consiste à effectuer un premier repérage des besoins fonctionnels et opérationnels. Elle prépare les activités plus formelles de capture des besoins fonctionnels et de capture des besoins techniques.

2. Présentation du projet

Afin d'identifier au mieux les besoins de l'application et de répondre aux attentes des utilisateurs, nous avons effectué plusieurs recherches sur les solutions open source dédié à la supervision des réseaux entre autres le logiciel *MRTG* (Multi Router Traffic Grapher) utilisé au sein d'*Algérie Télécom Satellite* avant l'aboutissement de notre application.

Ce logiciel utilise le protocole *SNMP* (vu dans le chapitre 2) pour collecter des données sur l'état du réseau (le trafic sur une interface, la température d'un équipement, l'espace disque disponible sur un serveur, ...) et les place dans un fichier journal « .log ». A partir de ces fichiers journaux, *MRTG* génère des graphiques (au format PNG) et des pages web (au format html) pour afficher les graphiques créés. Il présente l'avantage malgré toutes les données qu'il peut collecter d'être très performant et donc de ne pas influencer sur le trafic d'un réseau [Web 11]. Malgré tout, *MRTG* possède des inconvénients non négligeables :

- ✘ Il passe beaucoup de temps à générer des pages html
- ✘ Il ne peut tracer que des graphiques à deux variables
- ✘ Il ne gère pas les valeurs négatives et il ne conserve pas les données (fichiers « .log » de taille constante)
- ✘ Il ne présente aucun outil pour l'interprétation des informations de gestions récupérées
- ✘ Il ne gère pas les incidents qui peuvent se produire sur le réseau

L'ensemble de ces inconvénients a motivé *Algérie Télécom Satellite* à concevoir sa propre solution pour remédier aux lacunes du *MRTG* et répondre à ses exigences en terme de supervision des équipements relatives aux réseaux VSAT de ses clients.

3. Recueil des besoins fonctionnels et non fonctionnels

a) Exigences fonctionnels :

Un premier tour d'horizon des besoins exprimés par les employés de l'entreprise et des fonctionnalités offertes par les logiciels de supervision réseau gratuit nous a permis d'établir le cahier des charges préliminaire suivant :

- ✘ Authentification pour accéder aux différentes fonctionnalités du système, selon le profil de l'utilisateur
- ✘ Contrôler tout les équipements du réseau simultanément
- ✘ Récupérer les données de gestion des équipements de réseaux en utilisant le protocole SNMP et les stocker dans des fichiers journaux
- ✘ Interpréter les données contenues dans les fichiers journaux
- ✘ Générer des événements qui reflètent l'état des équipements de notre réseau et les stocker dans des fichiers journaux
- ✘ Produire des alertes en cas d'erreur sur le réseau
- ✘ Fournir les détails relatifs aux alertes
- ✘ Communication éventuelle entre le service de supervision et le service qui se chargent du traitement des alertes par le biais d'un client de messagerie
- ✘ Consulter l'historique des alertes relatives aux équipements et leurs composants
- ✘ Mise à jour de la liste des équipements et de leurs composants à contrôler

b) Exigences non fonctionnels :

- ✘ Réduire au maximum le temps de chargement des données
- ✘ Ergonomie et convivialité du produit
- ✘ Facilité d'utilisation

Une fois ce premier recueil de besoins effectué, la description du contexte du système peut commencer. Elle consiste en trois activités successives :

- ✓ L'identification des acteurs,
- ✓ L'identification des messages,
- ✓ La réalisation des diagrammes de contexte.

4. Identification des acteurs du système

Les acteurs qui interagissent directement avec notre système sont les suivants :

- ✓ **Superviseur** : c'est la personne chargée de suivre l'activité et l'état du réseau VSAT au sein d'ATS
- ✓ **Client** : il représente les entreprises étatiques ou privées qui bénéficient du service VSAT proposé par ATS

Remarque :

Le client n'accède qu'aux informations relatives à ses équipements à l'opposé du superviseur qui a droit de consulter toutes les informations concernant l'ensemble des équipements des clients d'ATS

5. Identification des messages

Dans ce qui suit nous définissons pour chaque acteur, les messages qui déclenchent un comportement du système attendu par l'acteur dans le cadre de son activité et les messages émis par le système à l'intention d'un acteur particulier, et qui portent une information utilisée par ce destinataire.

Messages émis par le système	Destinataire
Liste des équipements et leurs composants à superviser	Superviseur/Client
Liste des alertes générées par un équipement donné	Superviseur/Client
Liste des alertes générées par un composant donné	Superviseur/Client
Fiche des détails concernant une alerte	Superviseur/Client
Fiche des détails concernant un composant	Superviseur/Client
Historique des alertes	Superviseur
Rapport d'alerte	Superviseur

Tableau IV.1 : les messages émis par le système

Messages reçus par le système	Emetteur
Demande d'authentification	Superviseur/Client
Mise à jour de la liste des équipements et composants à superviser	Superviseur
Consulter les alertes générées par un équipement donné	Superviseur/Client
Consulter les alertes générées par un composant donné	Superviseur/Client
Demande de détail concernant une alerte donnée	Superviseur/Client
Demande de détail concernant un composant donnée	Superviseur/Client
Consulter l'historique des alertes générées par un composant donné	Superviseur
Demande d'envoi du rapport d'alerte	Superviseur

Tableau IV.2 : Messages reçus par le système

6. Modélisation du contexte

Tous les messages (système ↔ acteurs) identifiés précédemment peuvent être représentés de façon synthétique sur un diagramme, que l'on peut qualifier de diagramme de contexte dynamique.

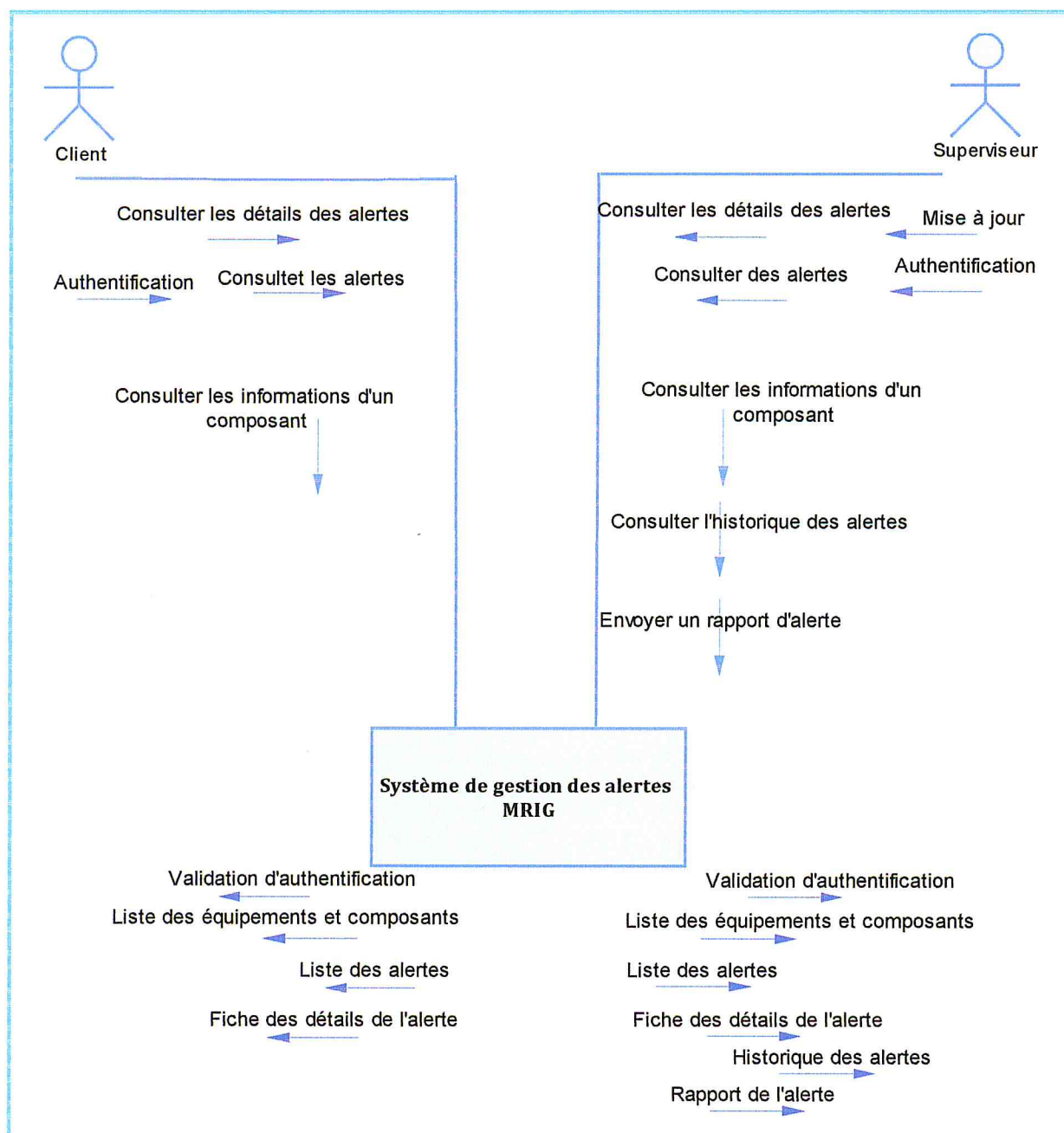


Figure IV.1 : Diagramme de contexte du système de gestion des alertes

7. Conclusion

L'étude préliminaire nous a servi à poser les bases de la capture des besoins du système à réaliser, les différentes fonctionnalités à mettre en œuvre seront détaillées dans le chapitre suivant.

Chapitre V: Etude Fonctionnelle

1. Introduction

Ce chapitre a pour but de compléter la capture des besoins fonctionnels ébauchée durant l'étude préliminaire. La technique des cas d'utilisation est la pierre angulaire de cette étape. Elle va nous permettre de préciser l'étude du contexte fonctionnel du système, en décrivant les différentes façons qu'auront les acteurs d'utiliser le futur système. Dans ce cadre nous allons :

- ✓ Identifier les cas d'utilisation du système par ses acteurs,
- ✓ Décrire les cas d'utilisation,
- ✓ Identifier les classes candidates du modèle d'analyse.

2. Capture des besoins fonctionnels

2.1. Identification des cas d'utilisation

Afin d'identifier les cas d'utilisation, on se servira avec profit du modèle de contexte présenté dans l'étude préliminaire, cela en considérant l'intention fonctionnelle de l'acteur par rapport au système dans le cadre de l'émission ou de la réception de chaque message et en regroupant les intentions fonctionnelles en unités cohérentes [ROQ2003].

Le résultat de cette étape est présenté dans le tableau ci-dessous qui montre le lien entre les cas d'utilisation identifiés, les acteurs et les messages provenant du contexte.

Cas d'utilisation	Acteurs	Messages émis /reçus par les acteurs
Authentification	Superviseur / Client	<u>Emet</u> : demande d'authentification <u>Reçoit</u> : l'accès a l'espace personnel.
Consulter la liste des alertes	Superviseur / Client	<u>Emet</u> : sélectionner un équipement / composant <u>Reçoit</u> : liste des alertes pour l'équipement/composant

Consulter les détails d'une alerte	Superviseur / Client	<u>Emet</u> : demande de détail <u>Reçoit</u> : fiche des détails.
Charger l'historique des alertes	Superviseur	<u>Emet</u> : sélectionner un composant <u>Reçoit</u> : historique des alertes.
Consulter les informations d'un composant	Superviseur / Client	<u>Emet</u> : sélectionner un composant <u>Reçoit</u> : fiche des détails.
Mettre à jour la liste des équipements et leurs composants	Superviseur / Client	
Envoyer un rapport d'alerte	Superviseur	<u>Emet</u> : envoyer un rapport <u>Reçoit</u> : rapport envoyé.

Tableau V.1 : cas d'utilisation du système de gestion des alertes

2.2. Les diagrammes des cas d'utilisation

✧ L'accès au système

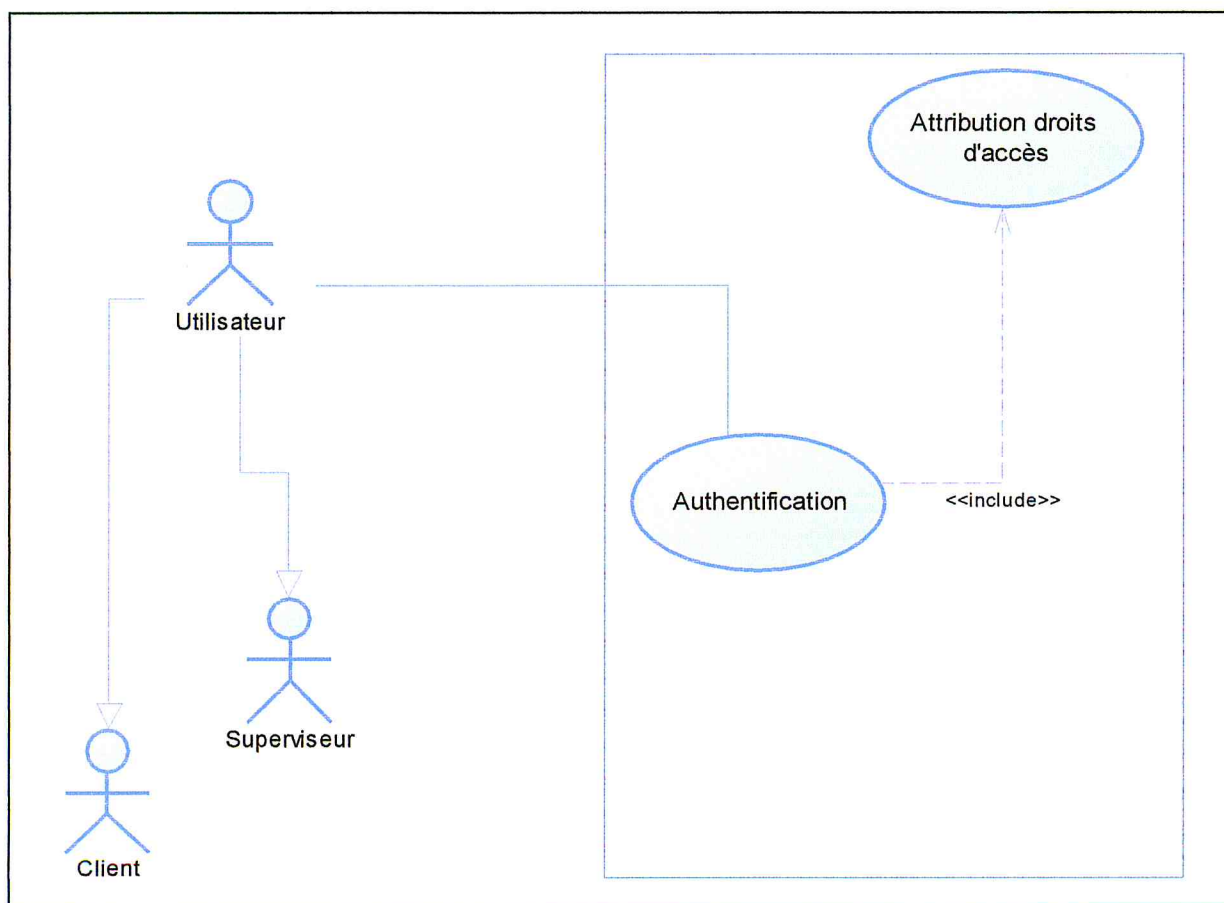


Figure V.1: diagramme de cas d'utilisations « accès au système »

✂ Description des cas d'utilisation

Un cas d'utilisation représente un ensemble de séquences d'interactions entre le système et ses acteurs. Pour décrire la dynamique du cas d'utilisation, le plus naturel consiste à recenser toutes les interactions de façon textuelle. La fiche de description textuelle d'un cas d'utilisation n'est pas normalisée par UML [ROQ2003]. Nous préconisons pour notre part la structuration suivante :

- ✓ **Un sommaire d'identification** : décrivant les objectifs du cas d'utilisation
- ✓ **Description des enchaînements** : représentant les interactions du cas d'utilisation

✂ Cas d'utilisation « Authentification »

✓ *Sommaire d'identification*

Titre : Authentification

But : l'accès à l'espace personnel

Résumé : s'identifier afin d'accéder aux fonctionnalités du système dédié à l'utilisateur

Acteurs : Superviseur, Client

✓ *Description des enchaînements*

Pré-conditions :

L'acteur doit être inscrit.

Le compte utilisateur doit être activé.

Scénario :

Ce cas d'utilisation commence lorsqu'un utilisateur demande à se connecter au système.

Enchaînement 1 : Accès à l'espace personnel

- L'acteur saisie les informations suivantes :
 - ✓ Nom d'utilisateur
 - ✓ Mot de passe
- Le système vérifie la validité des informations saisies et affecte ensuite les droits d'accès à l'utilisateur
- Le système présente l'espace personnel à l'acteur.

[Exception 1 : champValide] : le système peut déclencher une exception sur la validité d'un champ saisi.

Exception :

[Exception1 : champValide] : le système envoie un message d'erreur d'authentification : mot de passe ou nom d'utilisateur incorrecte.

Post-condition :

L'authentification est effectuée avec succès.

✘ **Processus de gestion des alertes :**

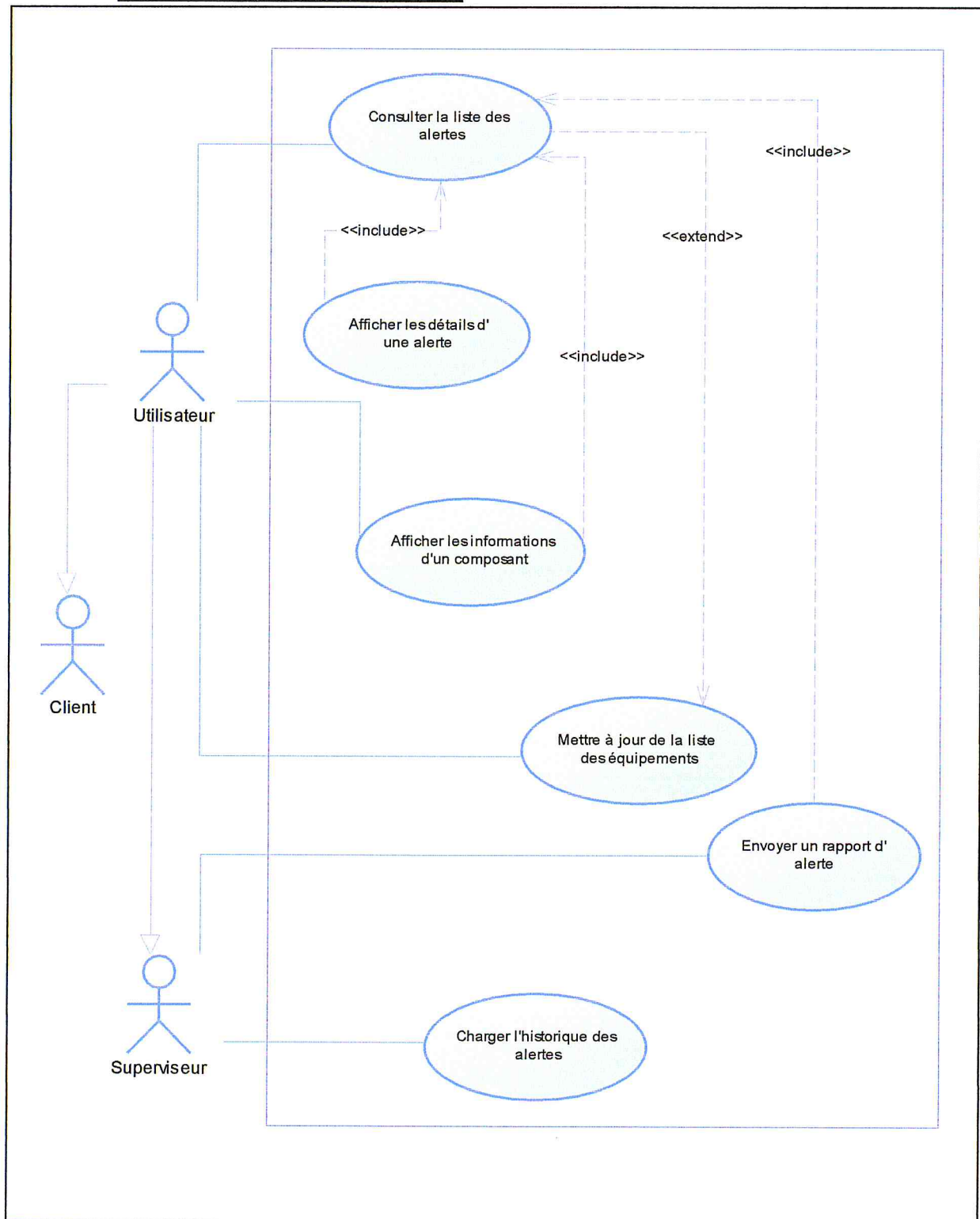


Figure V.2 : diagramme de cas d'utilisations « Processus de gestion des alertes »

✂ Cas d'utilisation « Consulter la liste des alertes »

✓ *Sommaire d'identification*

Titre : Consulter la liste des alertes

But : Afficher les alertes produites éventuellement pour chaque équipement et ses composants

Résumé : Rechercher et sélectionner un équipement pour l'affichage des alertes générées par celui-ci en temps réel

Acteurs : Superviseur, Client

✓ *Description des enchainements*

Pré-conditions :

L'acteur doit être authentifié

Il doit exister au moins un équipement ou un composant à contrôler

Scénario :

Ce cas d'utilisation commence lorsqu'un utilisateur demande au système d'afficher la liste des alertes générées par un équipement ou un composant donné en temps réel.

Enchainement 1 : Mettre à jour la liste des équipements et leurs composants

L'acteur demande au système de mettre à jour la liste des équipements et leurs composants, dans le cas où une opération d'ajout d'un nouvel équipement ou composant a lieu au même temps qu'une opération de consultation des alertes.

Enchainement 2 : Afficher la liste des alertes pour un équipement

- L'acteur (*superviseur*) sélectionne le nom d'un client
- Le système affiche la liste des équipements appartenant à ce client
- L'acteur choisit l'équipement dont il souhaite consulter les alertes
- Le système affiche la liste des alertes produites par cet équipement

Dans le cas où l'acteur est un *Client*, le système n'affiche que les équipements dont il est le propriétaire.

Enchainement 3 : Trier la liste des alertes

- L'acteur demande à trier la liste des alertes selon un des critères suivants :
 - ✓ L'adresse IP de l'équipement
 - ✓ Le numéro du composant de l'équipement
 - ✓ La date de l'alerte
 - ✓ Le type de l'alerte

Enchainement 4 : Afficher les détails d'une alerte

- L'acteur sélectionne une alerte est demande au système d'afficher plus de détails la concernant
- Le système affiche des informations suivantes :
 - ✓ Une brève description de la nature de l'alerte
 - ✓ Les causes potentielles qui l'ont généré

Enchaînement alternatif :

Enchaînement alternatif (a) : Afficher la liste des alertes pour un composant

- L'acteur (*superviseur*) sélectionne le nom d'un client
- Le système affiche la liste des équipements et leurs composants appartenant à ce client
- L'acteur choisi l'équipement et son composant dont il souhaite consulter les alertes
- Le système affiche la liste des alertes produites par ce composant

Post-condition :

La consultation des alertes est effectuée avec succès

✂ *Cas d'utilisation « Consulter les informations d'un composant »*

✓ *Sommaire d'identification*

Titre : Consulter les informations d'un composant

But : Afficher plus de détails concernant un composant qui produit une alerte

Résumé : Pour chaque composant d'un équipement supervisé, le système fournit une fiche descriptive

Acteurs : Superviseur, Client

✓ *Description des enchaînements*

Pré-conditions :

L'acteur doit être authentifié

Il doit exister au moins une alerte générée par le composant

Scénario :

Ce cas d'utilisation commence lorsqu'un utilisateur demande au système d'afficher les détails d'un composant produisant une alerte.

Enchaînement :

- Le système affiche la liste des alertes produites par un équipement
- L'acteur sélectionne une alerte
- Le système affiche une description du composant qui est à l'origine de cette alerte

Post-condition :

Les informations concernant le composant sont affichées avec succès

✂ Cas d'utilisation «Charger l'historique des alertes »

✓ *Sommaire d'identification*

Titre : Charger l'historique des alertes

But : Afficher l'historique des alertes produites par un composant donné

Résumé : Pour chaque composant d'un équipement supervisé, l'acteur est en mesure de consulter l'historique des alertes produites par celui-ci durant une période donnée

Acteurs : Superviseur

✓ *Description des enchainements*

Pré-conditions :

L'acteur doit être authentifié

Il doit exister au moins une alerte générée par le composant

Scénario :

Ce cas d'utilisation commence lorsqu'un utilisateur demande au système de charger l'historique des alertes générées par un composant donné.

Enchainement :

- Le système affiche la liste des alertes produites par un composant en temps réel
- L'acteur demande au système d'afficher l'historique des alertes générées par ce même composant
- Le système demande à l'acteur de spécifier la période pour laquelle il souhaite consulter l'historique des alertes
- L'acteur saisit deux dates entre lesquelles les éventuelles alertes qu'il veut afficher sont produites
- Le système charge l'historique et l'affiche

[Exception 1 : DateValide] : le système peut déclencher une exception sur la validité d'une date saisie.

[Exception2 : champObligatoire] : Le système peut déclencher une exception sur un champ obligatoire.

Exception :

[Exception1 : DateValide] : le système envoie un message d'erreur à l'acteur l'informant que le champ date n'est pas valide.

[Exception2 : champObligatoire] : un message d'erreur est affiché par le système avisant l'acteur qu'il doit saisir le champ date vide.

Post-condition :

L'historique des alertes du composant est chargé et affiché avec succès

✂ Cas d'utilisation «Envoyer un rapport d'alerte »

✓ *Sommaire d'identification*

Titre : Envoyer un rapport d'alerte

But : Envoyer un rapport d'alerte aux services chargés du traitement des incidents produits sur les équipements VSAT

Résumé : Après la consultation des alertes, le superviseur peut juger nécessaire de prévenir les services chargés de la maintenance des équipements VSAT que ce soit sur le site client ou bien à ATS des alertes, pour que ces derniers interviennent

Acteurs : Superviseur

✓ *Description des enchainements*

Pré-conditions :

L'acteur doit être authentifié

Il doit exister au moins une alerte générée par un composant

Scénario :

Ce cas d'utilisation commence lorsque le superviseur demande au système d'envoyer un rapport d'alerte

Enchainement :

- Le système affiche la liste des alertes produites par un équipement en temps réel
- Le superviseur évalue les alertes une par une et demande au système d'envoyer un rapport d'alerte à chaque fois qu'il le juge utile
- Le système affiche le rapport d'alerte
- Le superviseur choisit ou pas de joindre le rapport avec sa propre observation et saisit l'adresse du destinataire
- Le superviseur demande au système d'envoyer le message
- Le système affiche un message de confirmation de l'envoi du message

Post-condition :

L'envoi du rapport est effectué avec succès.

3. Analyse

A la fin de l'analyse des besoins, nous avons obtenu un découpage fonctionnel exprimé à travers les cas d'utilisation. Pour passer à l'analyse, nous allons changer radicalement l'organisation du modèle et nous fonder sur les principes de l'approche orientée objet [ROQ2003]. À cet effet, nous détaillerons le **modèle statique** de notre system en ce basant sur le **diagramme de classe**. Par la suite, le **modèle dynamique** sera traité par le biais du **diagramme de séquences**.

3.1. Développement du modèle statique

L'expression préliminaire des besoins donne lieu assez directement à une modélisation par les cas d'utilisation. Il s'agit là de descriptions fonctionnelles qui vont nous servir en particulier pour les tests de recette à la fin du projet, mais aussi de point d'entrée pour la description dynamique des scénarios d'exécution du futur système.

En revanche, la conception objet demande principalement une description structurelle, statique, du système à réaliser sous forme d'un ensemble de classes logicielles. Les premières classes candidates identifiées dans cette phase doivent être des concepts connus des utilisateurs du système, ce qu'on appelle couramment des objets métier [ROQ 2008].

Tout au long de notre développement, nous avons effectué un affinement sur ces classes avec l'ajout d'autres classes si cela s'est avéré nécessaire et l'affectation des attributs et associations entre les classes. Le résultat final de ce développement est présenté comme suit :

✓ Diagramme des classes

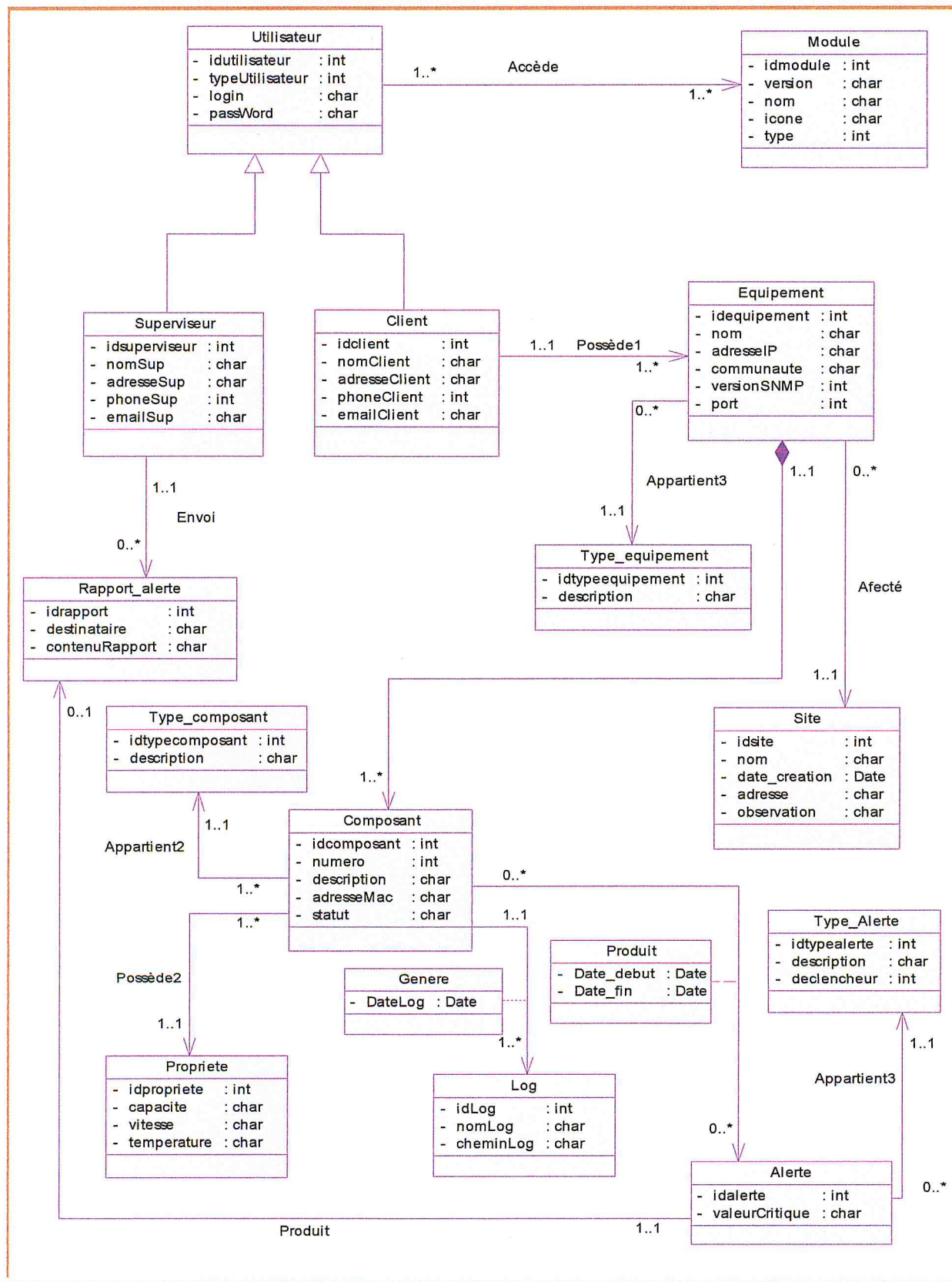


Figure V.3 : diagramme de classes du système de gestion des alertes

✓ Description des classes :

Classe	Attribut	Type	Désignations
<p>Utilisateur : Classe qui regroupe tout les utilisateurs du système</p>	<ul style="list-style-type: none"> • idutilisateur 	Entier	<ul style="list-style-type: none"> • Identifiant de l'utilisateur.
	<ul style="list-style-type: none"> • login 	Chaine de caractères	<ul style="list-style-type: none"> • Nom de l'utilisateur, pour l'authentification
	<ul style="list-style-type: none"> • PassWord 	Chaine de caractères	<ul style="list-style-type: none"> • Mot de passe de l'utilisateur
<p>Client : Classe héritant de la classe utilisateur, elle regroupe tous les clients inscrits au système</p>	<ul style="list-style-type: none"> • idclient 	Entier	<ul style="list-style-type: none"> • Identifiant du client.
	<ul style="list-style-type: none"> • nomClient 	Chaine de caractères	<ul style="list-style-type: none"> • Nom du client/entreprise
	<ul style="list-style-type: none"> • adresseClient 	Chaine de caractères	<ul style="list-style-type: none"> • Adresse du client /entreprise
	<ul style="list-style-type: none"> • villeClient 	Chaine de caractères	<ul style="list-style-type: none"> • Ville du client
	<ul style="list-style-type: none"> • phoneClient 	Entier	<ul style="list-style-type: none"> • téléphone de client.
	<ul style="list-style-type: none"> • emailClient 	Chaine de caractères	<ul style="list-style-type: none"> • Adresse email du client
<p>Superviseur : Classe héritant de la classe utilisateur, elle regroupe tous les superviseurs au sein d'ATS inscrits au système</p>	<ul style="list-style-type: none"> • idsuperviseur 	Entier	<ul style="list-style-type: none"> • Identifiant du superviseur
	<ul style="list-style-type: none"> • nomSup 	Chaine de caractères	<ul style="list-style-type: none"> • Nom du superviseur
	<ul style="list-style-type: none"> • adresseSup 	Chaine de caractères	<ul style="list-style-type: none"> • Adresse superviseur du
	<ul style="list-style-type: none"> • PhoneSup 	Entier	<ul style="list-style-type: none"> • Téléphone du superviseur
	<ul style="list-style-type: none"> • emailSup 	Chaine de caractères	<ul style="list-style-type: none"> • Adresse mail du superviseur
<p>Type Equipement : Classe regroupant tout les types d'équipement qui peuvent être superviser (pc, routeur, switch, firewall)</p>	<ul style="list-style-type: none"> • idtypeequipement 	Entier	<ul style="list-style-type: none"> • identifiant du type de l'équipement
	<ul style="list-style-type: none"> • NomTypeEquipement 	Chaine de caractères	<ul style="list-style-type: none"> • Nom du type de l'équipement

<p>Equipement: Classe regroupant tout les équipements appartenant aux réseaux VSAT et qui doivent être supervisés</p>	• Idequipement	Entier	• Identifiant de l'équipement
	• NomEquipement	Chaine de caractères	• Nom de l'équipement
	• Adresse_IP	Chaine de caractères	• Adresse IP de l'équipement
	• communaute	Chaine de caractères	• Type de la communauté SNMP
	• Version_SNMP	Entier	• La version SNMP installé sur l'équipement
	• port	Entier	• Numéro du port sur le quelle transitent les requêtes de l'agent SNMP
	• Cle_cryptage	Chaine de caractères	•
<p>Type Composant : Classe regroupant tout les types des composants des équipements du réseau qui doivent être supervisés (carte réseau, mémoire, processeur...)</p>	• idtypecomposant	Entier	• Identifiant du type du composant
	• typeComposant	Chaine de caractères	• Nom du type du composant
<p>Composant : Classe regroupant tout les composants des équipements à superviser</p>	• idcomposant	Entier	• Identifiant du composant
	• descriptionComposant	Chaine de caractères	• Le nom complet du composant
	• numComposant	Entier	• Le numéro du composant dans l'équipement
	• adresseMac	Chaine de caractères	• L'adresse mac du composant
	• statutComposant	Entier	• statut du composant
<p>Propriete : Classe qui indique les seuils à ne pas dépasser pour l'ensemble des valeurs des propriétés du composant</p>	• idpropriete	Entier	• Identifiant de propriété
	• capacite	Chaine de caractères	• La capacité maximale du composant
	• vitesse	Chaine de caractères	• La vitesse maximale du composant
	• temperature	Chaine de caractères	• La température maximale du composant

<p>Site : Classes regroupant les différents sites géographiques sur lesquels sont distribués les équipements</p>	• idsite	Entier	• Identifiant du site
	• nomSite	Chaîne de caractères	• Nom du site
	• dateCreation	Date	• Date de création du site
	• adresseSite	Chaîne de caractères	• Adresse du site
	• Observation	Chaîne de caractères	• Commentaire contenant des informations supplémentaires sur le site
<p>Module : Classe qui regroupe les différents modules qui constitue le système</p>	• idmodule	Entier	• Identifiant du module
	• VersionCourante	Chaîne de caractères	• Version du module
	• nomModule	Chaîne de caractères	• Nom du module
	• icone	Chaîne de caractères	• Url de l'icône qui représente le module
	• TypeModule	Entier	• Type du module
<p>Genere : Classe associative qui garde l'historique des dates de la génération d'un log pour un composant</p>	• DateLog	Date	• Date de création des logs
<p>Log : Classe regroupant tout les fichiers contenant les propriétés d'un composant</p>	• idLog	Entier	• Identificateur du log
	• nomLog	Chaîne de caractères	• Nom du log
	• cheminLog	Chaîne de caractères	• l'URL du log dans le serveur du système
<p>Produit : Classe associative qui garde l'historique des durés des alertes</p>	• Date_Debut	Date	• Date de l'apparition de l'alerte
	• Date_Fin	Date	• Date de la disparition de l'alerte
<p>Type Alerte : Classe regroupant les différents types d'alerte qui peuvent être produites par les différents types de composants</p>	• idtypealerte	Entier	• Identificateur du type d'alerte
	• description	Chaîne de caractères	• Brève description de l'alerte
	• declencheur	Chaîne de caractères	• Liste des principaux déclencheurs de l'alerte

<p>Alerte: Classe regroupant toutes les alertes produites par les différents composants</p>	<ul style="list-style-type: none"> • idalerte 	Entier	<ul style="list-style-type: none"> • Identificateur de l'alerte
	<ul style="list-style-type: none"> • ValeurCritique 	Float	<ul style="list-style-type: none"> • La valeur d'une propriété du composant qui à déclencher l'alerte
<p>RapportAlerte : Classe qui regroupe tout les rapports d'alerte émis par le superviseur</p>	<ul style="list-style-type: none"> • idrapport 	Entier	<ul style="list-style-type: none"> • Identificateur du rapport
	<ul style="list-style-type: none"> • destinataire 	Chaine de caractères	<ul style="list-style-type: none"> • Adresses des destinataires
	<ul style="list-style-type: none"> • contenuRapport 	Chaine de caractères	<ul style="list-style-type: none"> • Contenu du rapport

Tableau V.2 : tableau de descriptions des classes du système de gestion des alertes

3.2. Développement du modèle dynamique

Pour documenter les cas d'utilisation, la description textuelle est indispensable, car elle seule permet de communiquer facilement et précisément avec les utilisateurs. Elle est également l'occasion de s'entendre sur la terminologie employée, ainsi que d'identifier le contexte d'exécution de l'un ou de l'autre des enchaînements. En revanche, le texte présente des désavantages puisqu'il est difficile de montrer comment les enchaînements se succèdent.

Il est donc recommandé de compléter la description textuelle par un diagramme dynamique, qui apporte un niveau supérieur de formalisation, en montrant les différents messages échangés entre les instances de classes d'analyse [ROQ2008].

Notre choix s'est porté sur le diagramme de séquence car notre système se base principalement sur l'aspect chronologique des messages.

§ Les diagrammes de séquence

✕ Authentification :

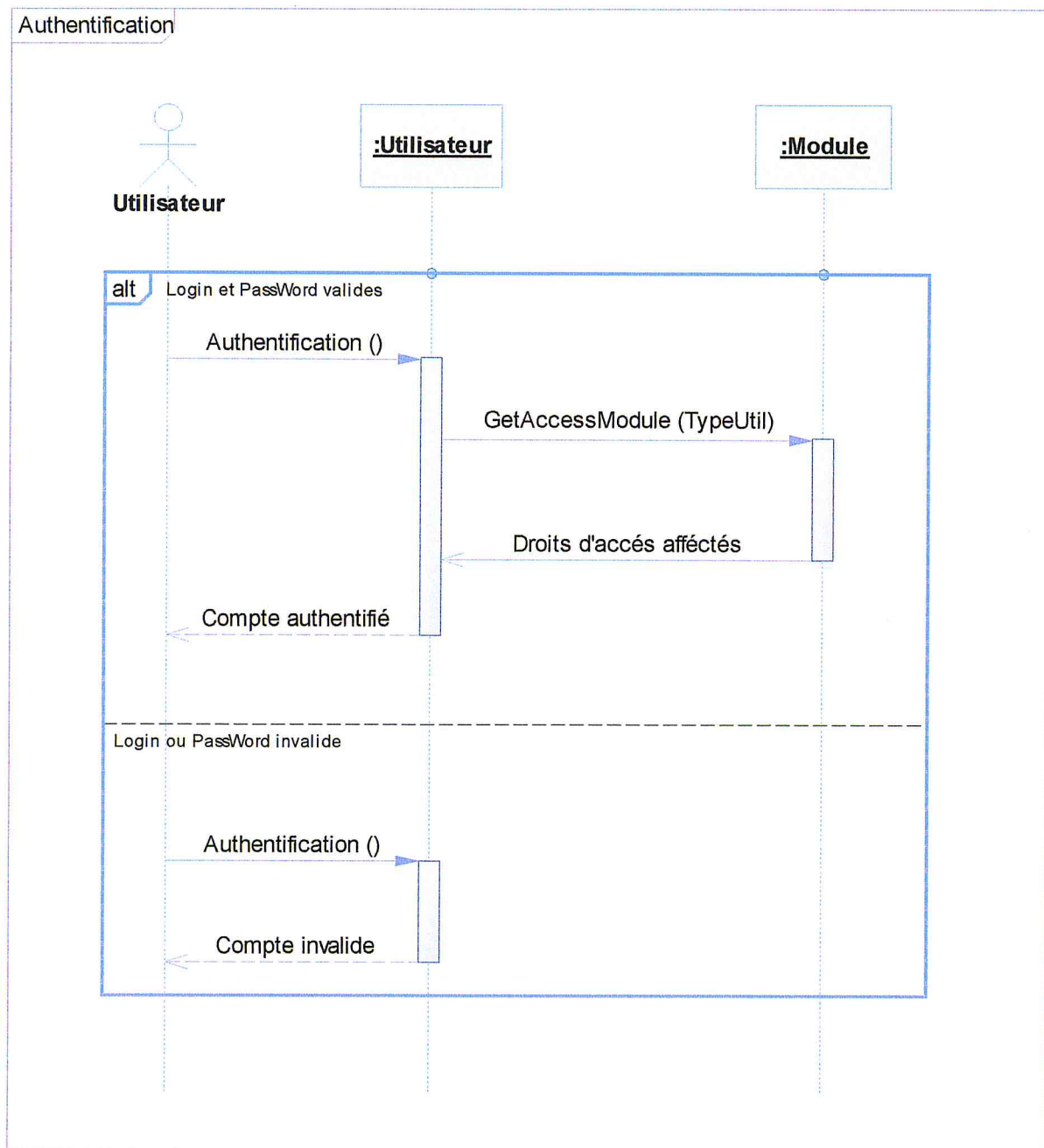


Figure V.4 : diagramme de séquences du cas d'utilisation « accès au système »

✕ Consulter les alertes :

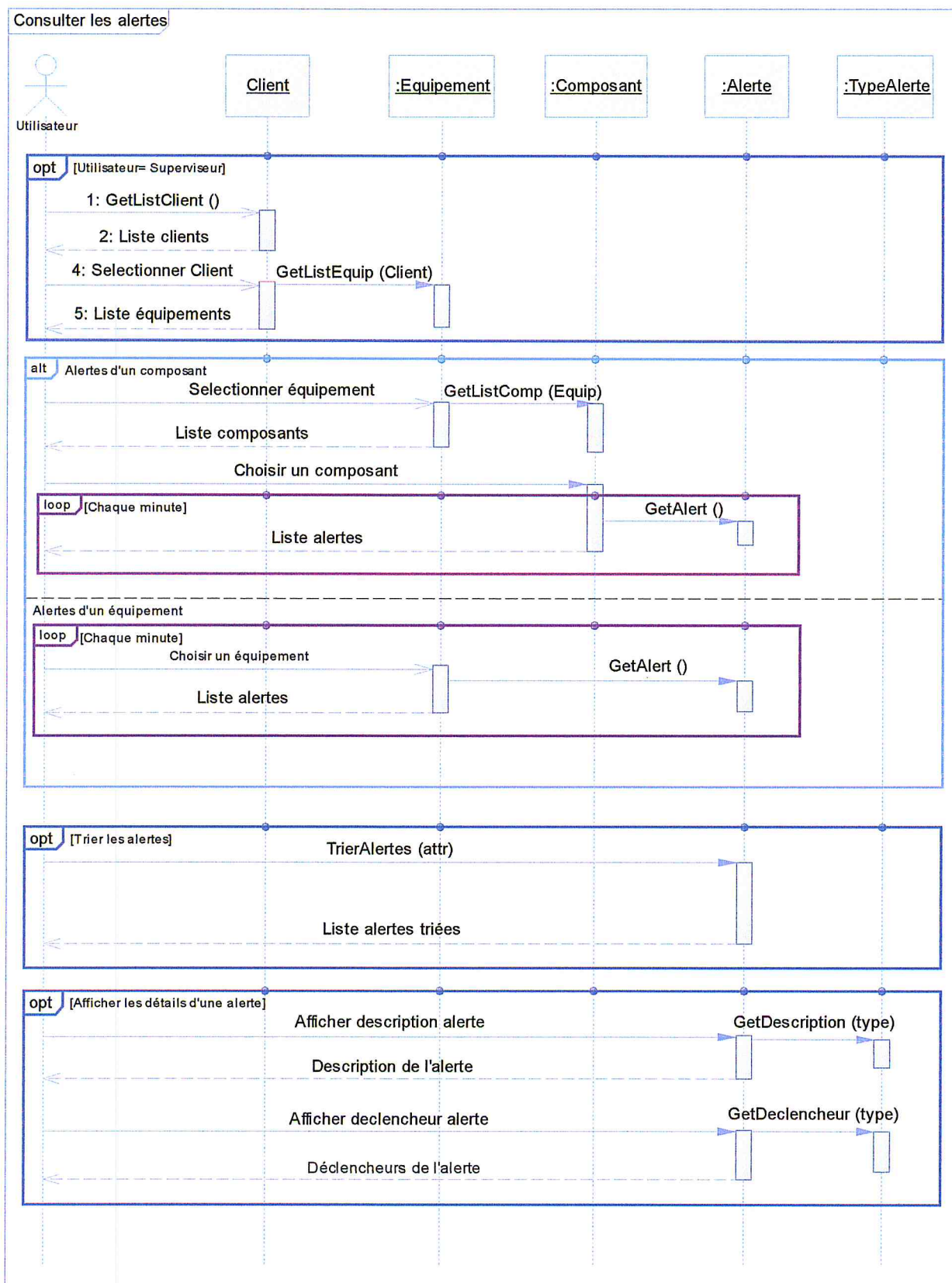


Figure V.5 : diagramme de séquences du cas d'utilisation « consulter les alertes »

✂ Consulter les détails d'un composant :

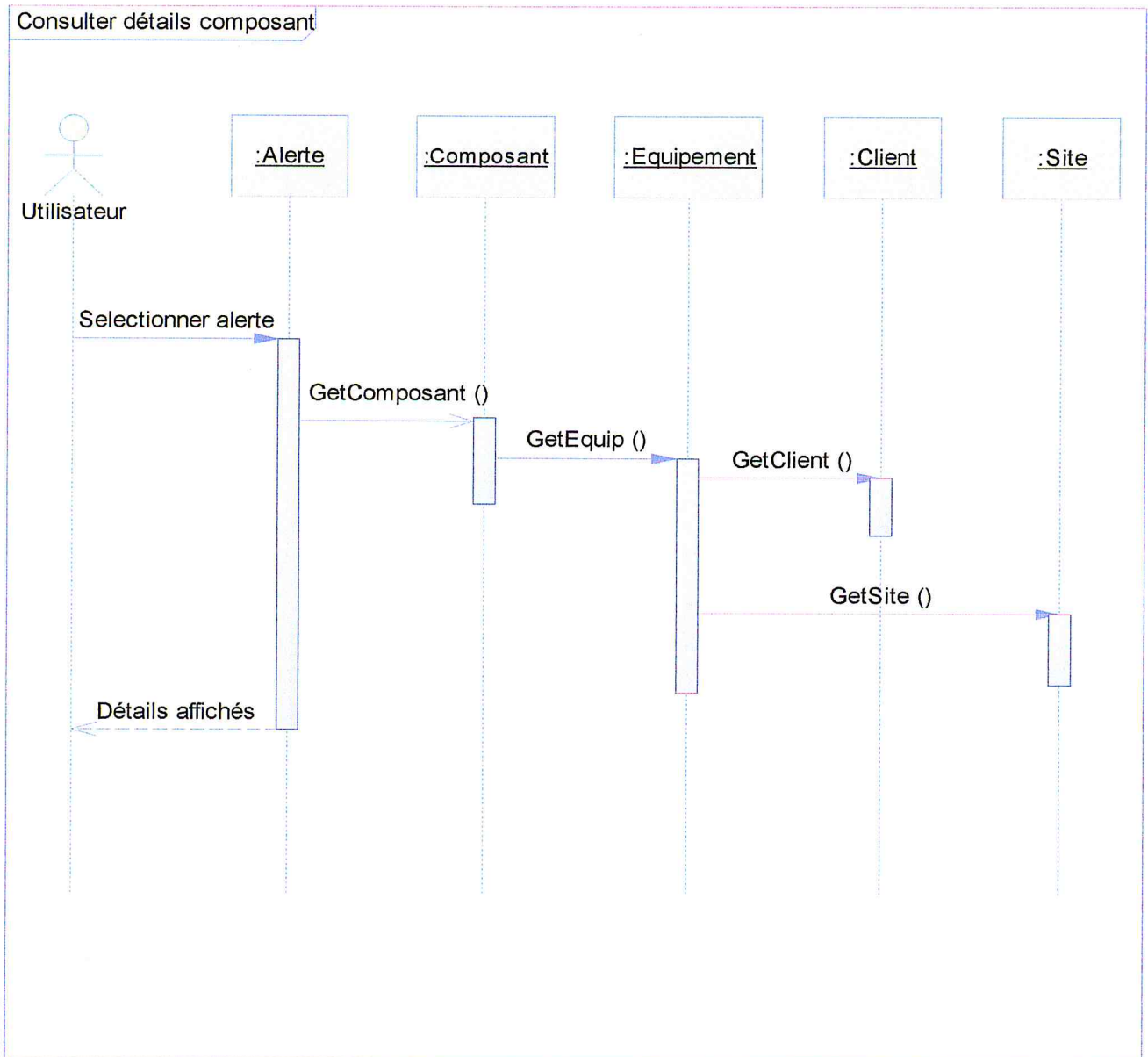


Figure V.6 : diagramme de séquences du cas d'utilisation «consulter les détails d'un composant»

✕ **Charger l'historique des alertes :**

Charger l'historique

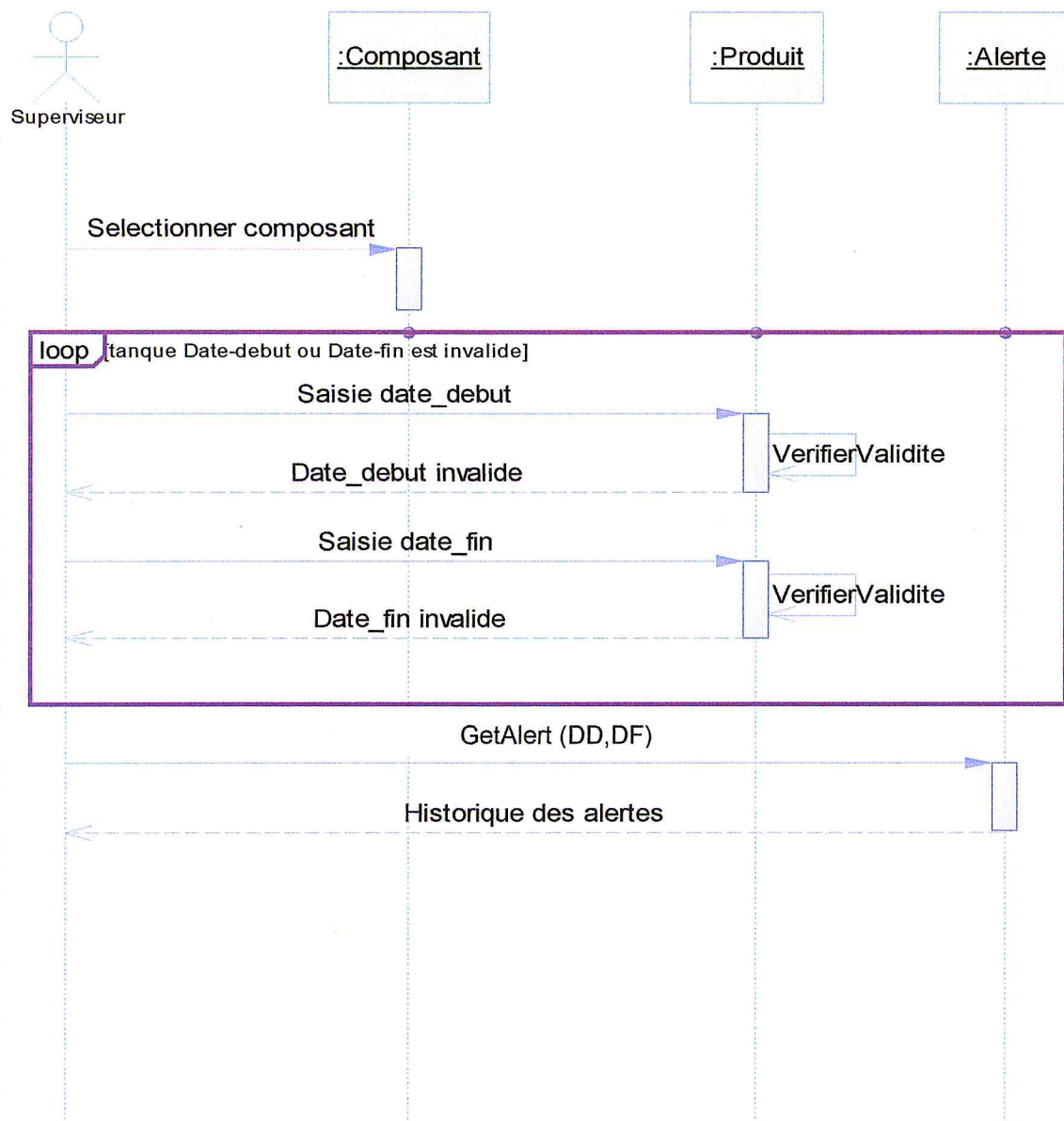


Figure V.6 : diagramme de séquences du cas d'utilisation « consulter l'historique des alertes »

✕ Envoyer un rapport d'alertes :

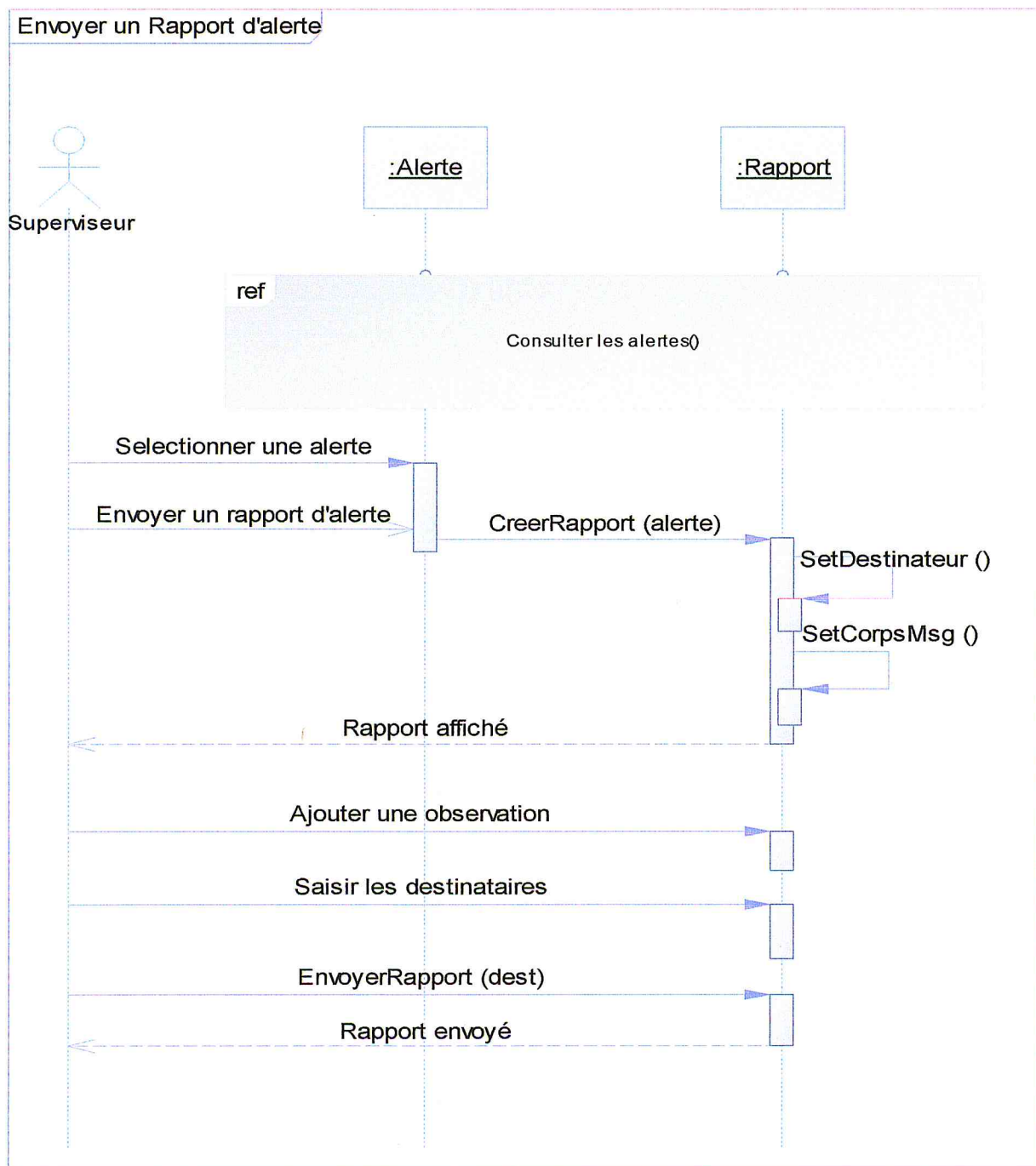


Figure V.7 : diagramme de séquences du cas d'utilisation «Envoyer un rapport d'alerte»

✦ **Générer les alertes:**

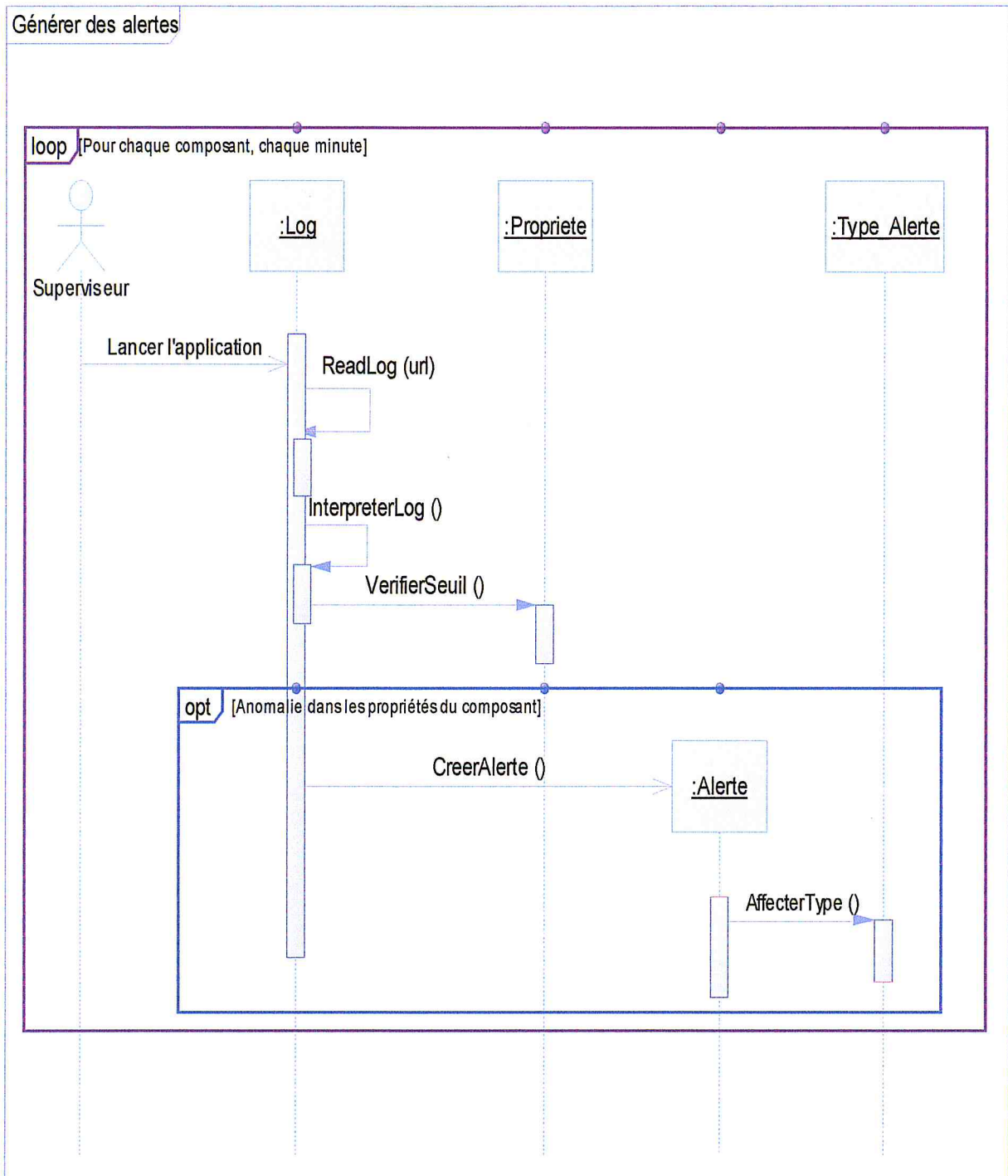


Figure V.5 : diagramme de séquences du cas d'utilisation «Générer les alertes»

Les correspondances entre les concepts du modèle statique et le modèle dynamique sont loin d'être triviales, car il s'agit bien de points de vue complémentaires, et non redondants. La confrontation entre notre diagramme de classe et diagrammes de séquences permet d'assembler les opérations effectuées sur les classes [ROQ2003].

Classe	Opération	Désignation
<u>Utilisateur</u>	Authentification ()	L'authentification de l'utilisateur par saisie de mot de passe et login
<u>Client</u>	GetListClient ()	Affiche la liste de tous les clients d'Algérie Télécom Satellite
<u>Module</u>	GetAccessModule (TypeUtil)	Attribut les droits d'accès aux différents modules du système selon le type de l'utilisateur
<u>Site</u>	GetSite (equip)	Retourne les propriétés d'un site donné
<u>Equipement</u>	GetListEquip (client)	Affiche la liste des équipements appartenant à un client donné
	GetEquip ()	Retourne les propriétés d'un équipement donné
<u>Composant</u>	GetListCmp (equip)	Affiche la liste de tous les composants d'un équipement donnée
	GetComposant ()	Récupère les propriétés d'un composant donné
<u>Propriete</u>	VerifierSeuil ()	Compare les différentes valeurs des propriétés d'un composant avec les seuils autorisés
<u>Log</u>	ReadLog (url)	Charge le document qui contient les informations sur l'état du composant, récupérées grâce au protocole SNMP et le lit
	InterpreterLog ()	Interprète chaque valeur d'une propriété du composant et détecte les éventuelles anomalies
<u>Alerte</u>	CreerAlerte ()	Cree une nouvelle alerte et définit ses attributs
	GetAlerte (DD,DF)	Affiche la liste des éventuelles alertes produites entre deux dates données
<u>Type Alerte</u>	AffecterType ()	Affecte un type à une alerte
	GetDescription (type)	Récupère la description d'un type d'alerte donné
	GetDeclencheur (type)	Récupère les déclencheurs d'un type d'alerte donné
<u>Rapport</u>	CreerRapport ()	Cree un rapport d'alerte concernant une alerte donnée

	EnvoyerRapport (dest)	Envoi le rapport à un ou plusieurs destinataires
<u>Produit</u>	VerifierValidite(Date)	Vérifie la validité des dates saisies

4. Conclusion

L'étude fonctionnelle nous a permis d'identifier les besoins exprimés selon le métier des utilisateurs, les structurer et les documenter avec UML en s'appuyant pour une large part sur la technique des cas d'utilisation et la modélisation objet du futur système en mode statique (diagramme de classe) et dynamique (diagramme de séquences).

Chapitre VI : Etude technique

1. Introduction

L'objectif de ce projet est la réalisation d'un ensemble d'interface web permettant la gestion des événements relatifs aux équipements du réseau VSAT. Il est alors nécessaire voire indispensable de traiter l'étude technique ou nous allons aborder la **capture des besoins techniques** qui décrira le choix des contraintes relatives à la configuration du réseau matériel et logiciel du système réalisé. En second, une **conception générique** s'impose pour développer la solution qui répond à ce qui a été retenu

2. Capture des besoins techniques :

La capture des besoins techniques couvre, par complémentarité avec celle des besoins fonctionnels, toutes les contraintes qui ne traitent ni de la description du métier des utilisateurs, ni de la description applicative [ROQ03]. Elle recense Les outils et les matériels à savoir les machines et réseaux, les progiciels à intégrer, et les outils retenus pour le développement. Cette étape se termine lorsque le niveau de description des cas d'utilisation techniques a permis l'identification des problèmes à résoudre.

2.1. Spécification de l'architecture à utiliser:

Le style d'architecture retenue pour notre système est l'architecture en tiers qui spécifie l'organisation des composants d'exploitation mis en œuvre pour réaliser le système. Un composant d'exploitation est une partie du système logiciel qui doit être connue, installée, déclarée et manipulée par les exploitants du système.

Un système client/serveur fait référence à au moins deux types de composants, qui sont les systèmes de base de données en serveur, et les applications qui en exploitent les données en client.

Dans le cadre des architectures d'entreprise, certains concepts et processus sont communs à plusieurs domaines d'activité. Cette caractéristique implique une synchronisation souvent complexe des données entre différents départements de l'entreprise. Le concept d'objet métier consiste à centraliser cette gestion afin d'en maîtriser la complexité. L'objet métier peut être perçu comme un modèle de composant d'exploitation qui s'insère dans le déploiement du

système d'entreprise [Eeles 98]. L'intégration des objets métier sous la forme de composants métier fait passer l'architecture client/serveur du 2-tiers au 3-tiers, car elle implique un nouveau type de composants d'exploitation qui s'insère entre les clients et les serveurs de données.

§ L'architecture 3-tiers

L'architecture 3-tiers (tiers signifie « partie » en anglais) est un modèle logique d'architecture applicative qui vise à séparer très nettement trois couches logicielles au sein d'une même application ou système, à modéliser et présenter cette application comme un empilement de trois couches dont le rôle est clairement défini [Web 12] :

- a. **Couche Présentation** : Elle correspond à la partie de l'application visible et interactive avec les utilisateurs. On parle d'Interface Homme Machine. En informatique, elle peut être réalisée par une application graphique ou textuelle. Elle peut aussi être représentée en *HTML* pour être exploitée par un navigateur web ou en *WML* (Wireless Markup Language) pour être utilisée par un téléphone portable.
- b. **Couche Métier** : correspond à la partie fonctionnelle de l'application, celle qui implémente la « logique », et qui décrit les opérations que l'application opère sur les données en fonction des requêtes des utilisateurs, effectuées au travers de la couche présentation.
- c. **Couche Accès aux données** : Elle consiste en la partie gérant l'accès aux gisements de données du système. Ces données peuvent être propres au système, ou gérées par un autre système. La couche métier n'a pas à s'adapter à ces deux cas, ils sont transparents pour elle, et elle accède aux données de manière uniforme.

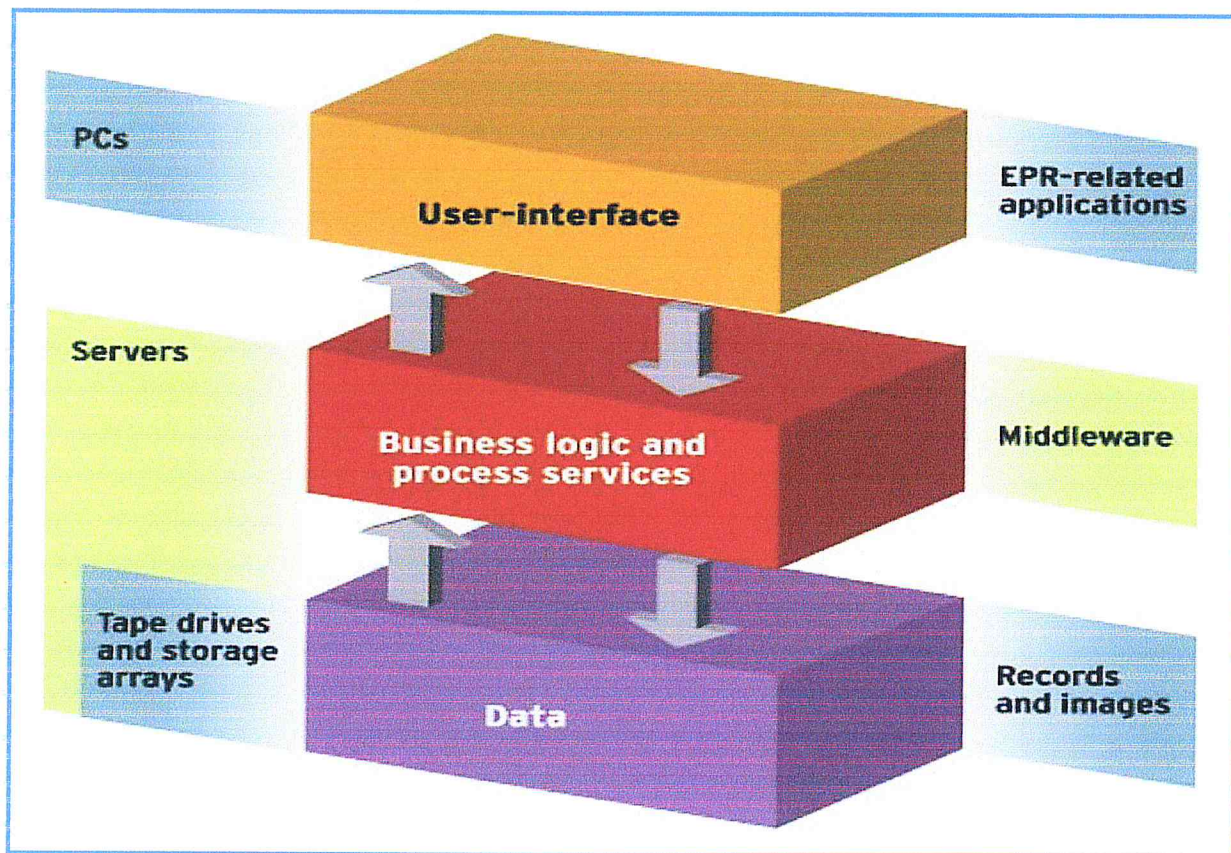


Figure VI.1 : Architecture 3-tiers

✂ Les avantages d'une architecture 3-tiers :

L'architecture à trois niveaux permet de :

- ✓ Faciliter la réutilisation au sein d'un système, puisque les composants métier correspondent à des concepts communs à différents métiers de l'entreprise [ROQ03].
- ✓ Fournir au système les moyens techniques qui lui permettent de garantir des temps de réponse constants, quel que soit le nombre d'utilisateurs connectés.
- ✓ D'un point de vue développement, la séparation qui existe entre le client, le serveur et le SGBD permet une spécialisation des développeurs sur chaque tiers de l'architecture [Web 13].
- ✓ Une plus grande flexibilité dans l'allocation des ressources, la portabilité du tiers serveur permet d'envisager une allocation et/ou modification dynamique au gré des besoins évolutifs au sein d'une entreprise.

2.2. *Élaboration du modèle de spécification logicielle:*

Nous nous intéressant dans cette partie aux fonctionnalités propre du système technique. Le modèle de spécification est abordé par le biais des cas d'utilisation mais d'une manière différente des cas d'utilisation fonctionnel. A cet effet, deux concepts sont introduit : l'exploitant et le cas d'utilisation technique [ROQ03].

✕ **L'exploitant**

L'exploitant est un acteur au sens d'UML, si ce n'est qu'il ne bénéficie que des fonctionnalités techniques du système.

✕ **Cas d'utilisation technique**

Un cas d'utilisation technique est destiné à l'exploitant. C'est une séquence d'actions produisant une valeur ajoutée opérationnelle ou purement technique.

2.2.1. *Les exploitants du système de gestion des alertes MRIG*

- ✓ L'utilisateur : regroupe tout les acteurs appartenant à la branche fonctionnelle.
- ✓ L'ingénieur d'exploitation : chargé de déployer et dépanner le système.

2.2.2. *Les diagramme des cas d'utilisation*

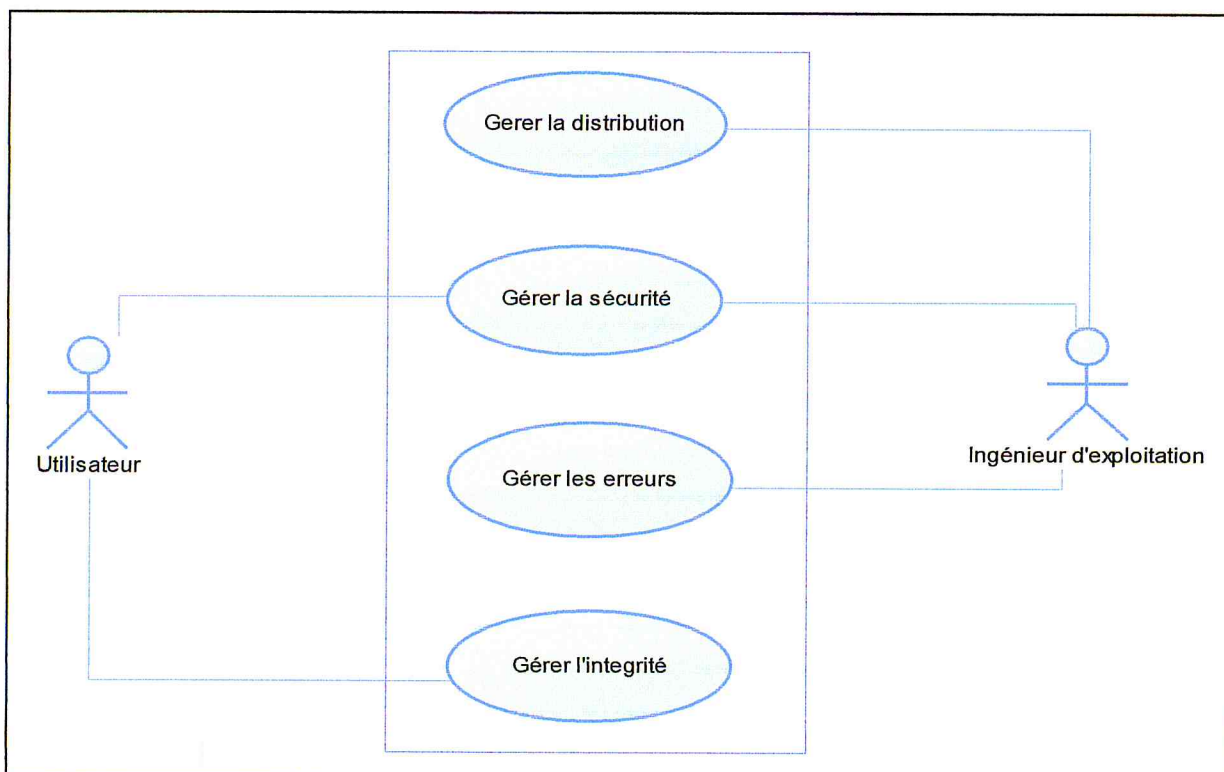


Figure VI.2 : diagramme des cas d'utilisation techniques

Cas d'utilisation	Description
Gérer la distribution	L'ingénieur d'exploitation doit assurer la gestion des charges au niveau du serveur. Ainsi, les temps de réponse du système ne s'en trouvent pas dégradés en fonction du nombre d'utilisateurs connectés
Gérer la sécurité	L'ingénieur d'exploitation ainsi que l'utilisateur sont soumis à des règles de sécurité. Dans un système client/serveur ces aspects recouvrent l'authentification, l'habilitation, le cryptage, la non-répudiation et l'audit.
Gérer les erreurs	Le système doit être exploitable, à ce titre, il faut qu'il soit en mesure de générer des traces et des alertes qui vont faciliter sa maintenance au sein du système informatique global de l'entreprise par l'ingénieur d'exploitation
Gérer l'intégrité	Plusieurs utilisateurs peuvent travailler en parallèle. L'intégrité est le mécanisme qui empêche la mise à jour simultanée d'une même entité par deux utilisateurs différents

Tableau VI.3 : tableau descriptive du diagramme des cas d'utilisation technique.

2.2.3. Les diagramme des cas d'utilisation

Le cas d'utilisation le plus pertinent à détailler est le cas d'utilisation « gérer la sécurité » car la réalisation du système se focalise principalement sur cet aspect.

✓ **Sommaire d'identification**

Titre : gérer la sécurité

But : garantir la sécurité du système.

Exploitation : ce cas d'utilisation est exploitable à tous les niveaux du système.

✓ **Description des enchainements**

Préconditions :

Aucune

Scenario nominal :

Ce cas d'utilisation est évoqué à chaque utilisation des services du système.

Enchainement 1 : assurer l'identité.

- Pour chaque session ouverte, le system garantie l'identité de l'utilisateur grâce au processus d'authentification

Enchainement2 : sécuriser les documents.

- Les offres doivent être transmises de manière à garantir leurs intégrités
- Les logs doivent être stockés dans le serveur de manière à garantir leurs authenticités

Enchainement3 : sécuriser l'accès aux données

- L'utilisateur n'accède qu'aux données qui lui sont permises (droit d'accès).

3. Conception générique :

La conception générique consiste à développer la solution qui répond aux spécifications techniques que nous avons présentées dans la phase de capture des besoins techniques [ROQ03].

Cette conception est qualifiée de générique car elle est entièrement indépendante des aspects fonctionnels spécifiés en branche fonctionnelle. La conception générique reste donc une activité de la branche technique qui nous permettra de voir tout les choix techniques qui seront utilisés pour la réalisation de notre système de gestion des alertes.

3.1. Les grands choix techniques :

3.1.1. Langage de programmation :

✂ Côté applicatif :

Le langage choisi pour la réalisation de notre application est **Java** par le biais des **JSP** (Java serveur page).

Java est un langage de programmation et une plate forme d'exécution, ce qui signifie que les programmes écrits en java peuvent être conçu pour fonctionner de la même manière sous Windows, sur des Macintosh et sous la plupart des versions d'Unix, afin de garantir la portabilité des applications développées en Java.

Les principales caractéristiques de java sont les suivants :

- ✓ Un langage de programmation orientée objet.
- ✓ Permet de développer des applications client/serveur en architecture distribuée.
- ✓ Java n'est pas exécuté, il est interprété par la machine virtuelle.
- ✓ Il a la capacité d'exécuter du code source extérieur de façon sécurisée.

✂ Java Server Page

Le Java Server Pages ou JSP est une technique basée sur Java qui permet aux développeurs web de générer dynamiquement du code HTML, XML ou tout autre type de page web [Web 14]. Il s'agit en réalité, d'un langage de script puissant, (Un langage interprété) exécuté du côté du serveur (au même titre que les scripts CGI, PHP, ASP,...) et non du côté client.

Les JSP sont intégrables au sein d'une page Web en HTML à l'aide de balises spéciales permettant au serveur Web de savoir que le code compris à l'intérieur de ces balises doit être interprété afin de renvoyer du code HTML au navigateur du client [Web 15].

Ainsi, les Java Server Pages s'inscrivent dans une architecture 3-tiers, cette dernière signifie qu'un serveur supportant les Java Server Pages peut servir d'intermédiaire (on parle généralement de serveur applicatif) entre le navigateur du client et une base de données (on parle généralement de serveur de données) en permettant un accès transparent à celle-ci.

✓ Le fonctionnement des Java Server Pages :

Une page utilisant les Java Server Pages est exécutée au moment ou le client émet une requête par un moteur de JSP, fonctionnant généralement avec un serveur Web ou un serveur applicatif.

Lorsqu'un utilisateur appelle une page JSP, le serveur Web appelle le moteur de JSP qui crée un code source Java à partir du script JSP, compile la classe afin de fournir un fichier compilé (d'extension .class), c'est-à-dire qu'il constitue en fait une *servlet* à partir du script JSP.

En réalité ce processus est un peu plus perfectionné: le moteur de JSP vérifie si la date du fichier JSP correspond à celle du fichier .class. Le moteur de JSP ne transforme et compile la classe que dans le cas où le script JSP a été mis à jour.

En effet, la plupart des technologies de pages actives (ASP, PHP, ...) reposent sur un code interprété, ce qui requiert beaucoup de ressources pour fournir la réponse HTTP. Etant donné que les JSP sont compilées (en fait il s'agit d'un bytecode) elles sont beaucoup plus rapides à l'exécution.

✓ *Caractéristiques des Java Server Pages*

Les JSP possèdent toutes les caractéristiques faisant la force de Java, ce qui fait qu'ils sont : multithread, Portables, orientées objet et sûres.

✕ *Les Servlets*

Les servlets sont des applications Java fonctionnant du côté serveur. Elles permettent de gérer des requêtes HTTP et de fournir au client une réponse HTTP dynamique (donc de créer des pages web dynamiques).

Les servlets ont de nombreux avantages par rapport aux autres technologies côté serveur. Tout d'abord, étant donné qu'il s'agit d'une technologie Java, les servlets fournissent un moyen d'améliorer les serveurs web sur n'importe quelle plateforme.

D'autre part les servlets sont beaucoup plus performantes que les scripts, car il s'agit de pseudo-code, chargé automatiquement lors du démarrage du serveur ou bien lors de la connexion du premier client. Les servlets sont donc actives (résidentes en mémoire) et prêtes à traiter les demandes des clients grâce à des threads, tandis qu'avec les langages de script traditionnels un nouveau processus est créé pour chaque requête HTTP. Cela permet donc une charge moins importante au niveau du processeur du serveur, ainsi que de prendre une place moins importante en mémoire.

L'un des principaux atouts des servlets est la réutilisation, permettant de créer des composants encapsulant des services similaires, afin de pouvoir les réutiliser dans des applications futures [Web13].

Enfin une servlet étant une application Java, peut utiliser toutes les API Java afin de communiquer avec des applications extérieures, se connecter à des bases de données, accéder aux entrées-sorties (fichiers par exemple), ...

✧ *Les java beans*

Les java beans sont des composants réutilisables introduits par le *JDK 1.1 (Java Development Kit)*. Ce sont des objets autonomes qui doivent pouvoir être facilement assemblés entre eux pour créer un programme.

Les Java beans proposés par *Sun* reposent bien sûr sur java et de fait en possèdent toutes les caractéristiques : indépendance de la plate-forme, taille réduite du composant, ...

La technologie java beans propose de simplifier et faciliter la création et l'utilisation de composants.

Les java beans possèdent plusieurs caractéristiques [Web11]:

- ✓ La persistance : elle permet grâce au mécanisme de sérialisation de sauvegarder l'état d'un bean pour le restaurer ainsi si on assemble plusieurs beans pour former une application, on peut la sauvegarder.
- ✓ La communication grâce à des événements qui utilise le modèle des écouteurs introduit par java 1.1
- ✓ L'introspection : ce mécanisme permet de découvrir de façon dynamique l'ensemble des éléments qui compose le bean (attributs, méthodes et événements) sans avoir le code source.
- ✓ La possibilité de paramétrer le composant : les données du paramétrage sont conservées dans des propriétés.

✧ *L'API JavaMail*

Le courrier électronique repose sur le concept du client/serveur. Ainsi, l'utilisation d'e mail requiert deux composants :

- ✓ Un client de mail (Mail User Agent : MUA) tel que Outlook, Messenger, Eudora, ...
- ✓ Un serveur de mail (Mail Transport Agent : MTA) tel que SendMail

Les clients de mail s'appuient sur un serveur de mail pour obtenir et envoyer des messages. Les échanges entre client et serveur sont normalisés par des protocoles particuliers.

JavaMail est une API qui permet d'utiliser le courrier électronique (e-mail) dans une application écrite en java (application cliente, applet, servlet, EJB, ...). Son but est d'être facile à utiliser, de fournir une souplesse qui permet de la faire évoluer et de rester le plus indépendant possible des protocoles utilisés [Web12].

Coté interface homme machine (IHM) :

✕ *JavaScript:*

Javascript est un langage de script orienté objet principalement utilisé dans les pages *HTML*. A l'opposé des langages serveurs (qui s'exécutent sur le site), Javascript est exécuté sur l'ordinateur de l'internaute par le navigateur lui-même. Ainsi, ce langage permet une interaction avec l'utilisateur en fonction de ses actions [Web 16]. Comme par exemple, réagir aisément à un changement de focalisation ou un clic de souris.

✕ *CSS (Cascading Style Sheets) :*

Le principe des feuilles de style consiste à regrouper dans un même document des caractéristiques de mise en forme associées à des groupes d'éléments [Web 17]. Il suffit de définir par un nom un ensemble de définitions et de caractéristiques de mise en forme, et de l'appeler pour l'appliquer à un texte.

Les feuilles de style ont été mises au point afin de compenser les manques du langage HTML en ce qui concerne la mise en page et la présentation. En effet, le HTML offre un certain nombre de balises permettant de mettre en page et de définir le style d'un texte, toutefois chaque élément possède son propre style, indépendamment des éléments qui l'entourent.

Les feuilles de style permettent notamment :

- ↳ d'obtenir une présentation homogène sur tout un site en faisant appel sur toutes les pages à une même définition de style ;
- ↳ de permettre le changement de l'aspect d'un site complet entier par la seule modification de quelques lignes ;
- ↳ une plus grande lisibilité du HTML, car les styles sont définis à part ;
- ↳ des chargements de page plus rapides, pour les mêmes raisons que précédemment ;
- ↳ un positionnement plus rigoureux des éléments.

✕ *ExtJS* : [Web 20]

ExtJS est une bibliothèque JavaScript servant à construire des applications web interactives ressemblant à des applications bureau Linux, Windows... Cette bibliothèque met à disposition des développeurs un ensemble de composants visuels d'une grande qualité comme des onglets, des arbres, des champs de formulaires... ExtJS réagit avec l'utilisateur au moyen d'un gestionnaire d'événements *Event Manager* qui répond à tout type de requête utilisateur comme le déplacement ou le redimensionnement d'une fenêtre, le clic sur un bouton...

ExtJS est une bibliothèque écrite en JavaScript, donc toute application réalisée avec cette librairie est exécutée côté client de ce fait elle communique avec le serveur en utilisant AJAX, ainsi, tout requête est traité en arrière plan sans obligation de rafraichissement de la page web ce qui constitue son principal avantage.

3.1.2. *Serveur d'application* :

✕ *GlassFish* :

Le serveur d'applications Glassfish est le premier serveur Open Source à implémenter totalement la norme *JEE 5 (Java Enterprise Edition)* développé par la société *Sun Microsystems* [LAF10]. Le mot d'ordre de ce serveur est rapidité et performance. Actuellement, GlassFish est le seul serveur totalement compatible avec Java EE 6, il est d'ailleurs utilisé comme serveur de référence par plusieurs sociétés comme la SNCF et RTL.

✕ *Avantages de GlassFish* :

Voyons ce qui a pu motiver le lancement d'un tel projet alors qu'il existe déjà plusieurs solutions éprouvées dans le domaine :

- ✓ Il surpasse les capacités de Tomcat par exemple qui ne propose qu'un conteneur de JSP/servlet sans le support des EJB entre autres.
- ✓ Le serveur d'application de SUN est disponible sous une double licence Common Development and Distribution License (CDDL) et GNU General Public License (GPL) avec exception du classpath.
- ✓ Il propose un logiciel serveur pour les applications Web et un développement rapide pour un coût moindre [LAF10].
- ✓ Le serveur GlassFish est également compatible et permet un fonctionnement avec le serveur Apache.

⌘ Caractéristiques :

- ✓ Une des plus spectaculaire est sans doute son temps de démarrage qui est de l'ordre des 3 secondes alors qu'il avoisine les 30 secondes voir beaucoup plus pour certains de ses concurrents.
- ✓ Glass Fish au moment de démarrer ne va charger que les modules minimum dont il a besoin et devra ensuite charger les autres au moment où ils seront nécessaires.
- ✓ le concept d'*edit-save-refresh* browser permet de modifier du code à chaud (par exemple le code d'une servlet) et de sauvegarder la modification. Il suffit de rafraichir le navigateur pour voir la modification prise en compte.

Les deux premiers points peuvent paraître mineurs car ils n'affectent évidemment pas l'utilisateur final de l'application mais ils prennent tout leurs sens quand on connaît le temps qu'il faut habituellement passé en phase de développement à attendre que le serveur redémarre.

Côté fonctionnalités intéressantes, Glass Fish propose aussi une interopérabilité accrue avec des services web .NET 3.0 et bien d'autres que SUN.

Glass Fish Enterprise Manager permet d'avoir à disposition des sondes SNMP, intégrable dans la majorité des outils de supervision. Il propose aussi son outil de statistiques, Performance Advisor, qui ouvre la voie à l'optimisation des performances de Glassfish avec son binôme Performance Monitor.

3.1.3. System de gestion de base de données (SGBD) :

ORACLE 10g :

Une base de données complète destinée aux petites et moyennes entreprises exigeant de bénéficier des performances, de la disponibilité et de la sécurité de la base de données leader à un tarif économique. Utilisable sur serveur en cluster ou non pouvant comporter jusqu'à quatre processeurs, la base de données Oracle 10g [Web 21] constitue la solution idéale pour développer et déployer des applications de base de données de façon rentable

Il propose des fonctionnalités qui permettent d'assurer [Web 18]:

- ✓ La définition et la manipulation des données.
- ✓ La cohérence des données.
- ✓ La confidentialité des données.

- ✓ L'intégrité des données.
- ✓ La sauvegarde et la restauration des données.
- ✓ La gestion des accès concurrents.

3.1.4. Environnement de développement :

Notre application est développée dans l'environnement **NetBeans**.

Netbeans est l'environnement de Développement Intégré (EDI) supporté par SUN. Il est particulièrement bien adapté pour le développement d'applications WEB. Il remplace l'IDE Java Studio Creator [Web 19].

C'est un IDE moderne offrant un éditeur avec des codes couleurs et un ensemble de signes, des modèles de projets multi-langage et de différents types (application indépendante, distribuée, plugin, mobiles, ...), le refactoring, l'éditeur graphique d'interfaces et de pages web pour supporter le programmeur dans son travail. Il permet d'accéder rapidement à la documentation détaillée, de naviguer dans les sources et de faire des recherches d'usage des classes, méthodes et propriétés. Netbeans indique à l'utilisateur les erreurs et fait des propositions pour y remédier. Un déboggeur permet l'exécution pas à pas. Un suivi des ressources utilisées (cpu, mémoire) par le logiciel développé peut être fait via un profiler. Un framework de test unitaire tel que Junit peut être utilisé.

L'EDI NetBeans fournit des outils pour construire tous les composants Java EE, ce qui inclut les Enterprise Java Beans (EJBs), les pages web, les servlets, et les services web. Il intègre le serveur d'application Glassfish, ce qui permet de facilement développer des EJB et de les déployer.

Pour faire des applications, des éditeurs visuels sont disponibles pour construire rapidement des applications web basées sur des standards utilisant Ajax, CSS, JavaScript, et JSF.

4. Diagramme de déploiement du système de gestion des alertes

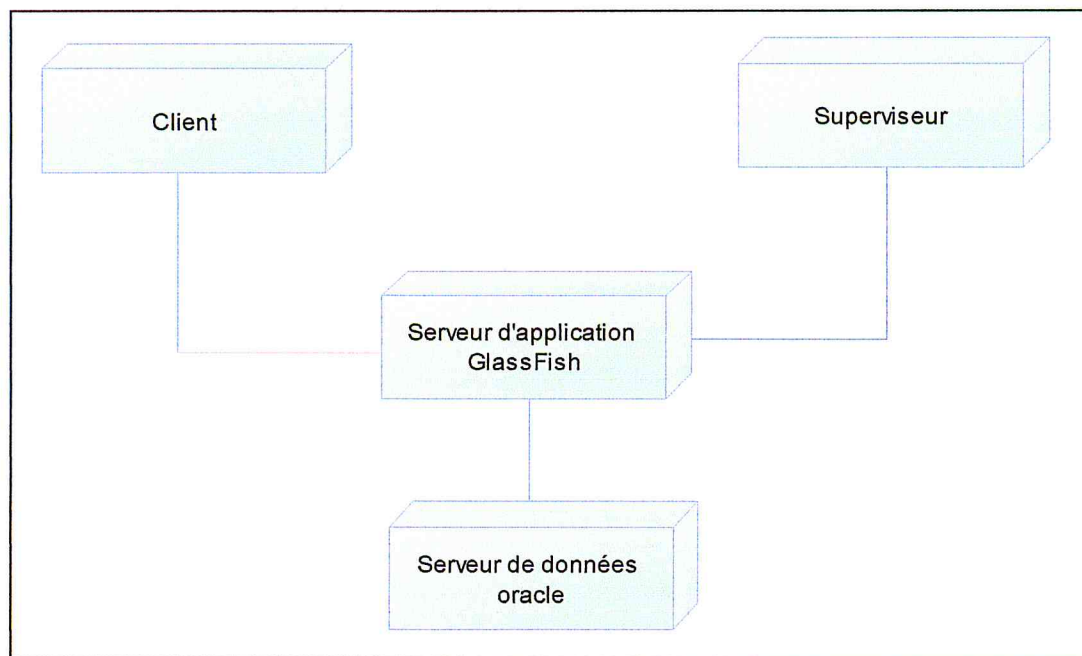


Figure VI.3 : Diagramme du déploiement du système de gestion des alertes

5. Conclusion

Dans ce chapitre nous avons essayé d'exposer l'étude technique, elle nous a permis de prendre en compte des contraintes techniques et logicielles pour la réalisation de notre système logiciel. Le chapitre qui suit présentera la fusion des études fonctionnelle et technique.

Chapitre VII : Conception et réalisation

1. Introduction

Après les études fonctionnelle et technique que nous avons réalisées, nous allons maintenant passer à la phase de conception. Le processus de conception permet de passer de l'analyse à la fabrication de la solution en langage objet. C'est à cette étape du processus 2TUP que s'effectue la fusion des études fonctionnelle et technique.

2. conception préliminaire :

La conception préliminaire est certainement l'étape la plus délicate du processus 2TUP car elle en représente le cœur, elle s'appuie sur les points de vue de spécification fonctionnelle et structurelle de l'analyse, mais également sur les frameworks de la conception technique.

La conception préliminaire peut être considérée comme la transformation du modèle d'analyse par projection des classes d'analyse sur les couches logicielles [ROQ2003].

2.1. Enumération des interfaces utilisateur

Si les composants communiquent par le biais de leurs interfaces, les applications, quant à elles, sont utilisables par le biais de leurs interfaces utilisateur ou **IHM** (*Interface Homme Machine*). Voici Une description d'une liste des vues avec leurs principales fonctions de notre application :

Vues IHM	Acteur	Description
Authentification de l'utilisateur	Superviseur Client	<ul style="list-style-type: none"> • Formulaire de demande de connexion, l'utilisateur doit remplir les champs suivants : <ul style="list-style-type: none"> ✓ Nom d'utilisateur ✓ Mot de passe puis demander à se connecter. • Le système renvoi une erreur si l'un des deux champs n'est pas valide

Afficher la liste des clients, des équipements et leurs composants	Superviseur	<ul style="list-style-type: none"> Page affichant la liste des clients d'ATS, de leurs équipements et leurs composants, elle propose également une option de recherche, ou le superviseur spécifie le nom du client ou de l'équipement qu'il souhaite afficher
Afficher la liste des équipements et leurs composants	Client	<ul style="list-style-type: none"> Page affichant la liste des équipements et leurs composants appartenant à un client donné
Afficher la liste des alertes par équipement ou composant	Superviseur Client	<ul style="list-style-type: none"> Tableau dynamique affichant la liste des alertes produites par un équipement ou un composant, la date de l'alerte et les valeurs critiques qui l'ont déclenché en temps réel Le tableau propose une fonction de tri selon l'une de ses colonnes du tableau
Afficher les détails d'une alerte	Superviseur Client	<ul style="list-style-type: none"> Fenêtre qui affiche une brève description de l'alerte et ses déclencheurs probables
Afficher les détails d'un composant	Superviseur Client	<ul style="list-style-type: none"> Une fenêtre affiche <ul style="list-style-type: none"> ✓ Nom du client ✓ Nom d'équipement ✓ Type de l'équipement ✓ Nom du composant ✓ Type du composant ✓ Adresse du site du composant qui a été à l'origine de l'alerte sélectionné dans la liste des alertes
Envoyer un rapport d'alerte	Superviseur	<ul style="list-style-type: none"> Une fenêtre s'affiche lorsque le superviseur choisit d'envoyer un rapport détaillant une alerte au service chargé de la traiter. Elle se compose de : Le rapport d'alerte contenant les informations suivantes : <ul style="list-style-type: none"> ✓ L'adresse IP de l'équipement ✓ Le numéro du composant ✓ Le motif de l'alerte ✓ La durée de l'alerte ✓ Les causes probables de l'alerte Une partie réservée pour l'ajout d'une éventuelle observation par le superviseur Un champ pour spécifier une ou plusieurs adresses de destination Champ « Objet » du rapport

<p>Charger l'historique des alertes</p>	<p>Superviseur</p>	<ul style="list-style-type: none"> • Le système renvoi une erreur si l'une des deux dates n'est pas valide Sinon : • 1^{re} étape : sélectionner un composant de la liste des composants • 2^{eme} étape : remplir les champs de la fenêtre « charger historique » <ul style="list-style-type: none"> ✓ Champ Date début : rempli par la date que le superviseur sélectionne sur le calendrier ✓ Champ Date fin : rempli par la date que le superviseur sélectionne sur le calendrier • 3eme étape : un tableau affiche toutes les alertes du composant sélectionné qui ont lieu entre les deux dates saisies.
---	--------------------	---

Tableau VII.1 : tableau descriptive des vues d'IHM de notre Application pour les utilisateurs

3. conception détaillée

A partir du modèle logique de conception et des interfaces, la méthode 2TUP-UML permet de définir les classes à implémenter. Cette activité consiste à :

- ↳ **Concevoir les classes :** consiste à transformer les classes métiers provenant de l'analyse en codage dans un langage de développement [ROQ2003].
- ↳ **Concevoir les attributs:** identifier les structures de données, les itérations et d'autres types complexes permettant de représenter les attributs d'analyse dans le langage utilisé.

Durant cette étape, la génération du schéma relationnel de la base de données du système est effectuée [ROQ2003].

3.1. Conception des classes

Les classes provenant de l'analyse sont conformes au langage d'implémentation choisit qui est JSP.

3.2. Conception des attributs

La conception des attributs consiste principalement à préciser le type des attributs identifiés dans la phase d'analyse. La visibilité des attributs ainsi que les modes d'accès. Par défaut tous les attributs sont privés (private). Chaque attribut est définit par deux méthodes «get» et «set» qui permettent respectivement de retourner et de renseigner la valeur d'un attribut.

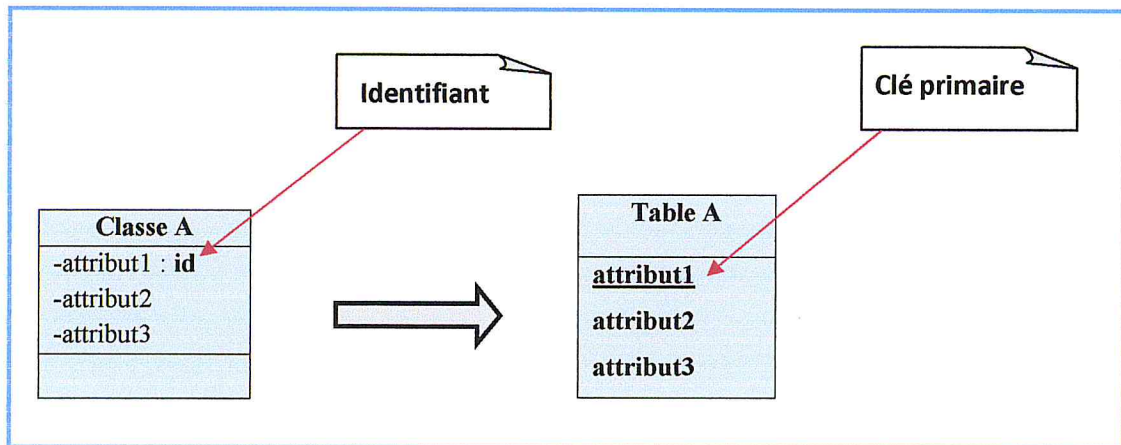
3.3. Règles de passage du modèle de conception au modèle relationnel :

L'utilisation d'un SGBDR impose un changement de représentation entre la structure des classes et la structure des données relationnelles [ROQ2003].

Dans ce qui suit nous décrivons les règles de passage du modèle de conception au modèle relationnel.

3.3.1. Transformation des classes (règle 1):

Le passage d'une classe à une table dans la base de données est montré dans le schéma ci-dessous. L'identifiant de la classe devient une clé primaire dans la table résultante [ROQ2005].



FigureVII.1 : Transformation d'une classe.

3.1.2. Transformation des relations :

3.1.2.1. Relation un-à-plusieurs (règle 2):

S'applique à toutes les relations comportant (x, 1) dans une extrémité de la relation (avec x appartient à {0,1}). Dans cette situation, on applique la règle de passage précédente. En plus, la clé primaire de la table B migre vers la table A et devient une clé étrangère.

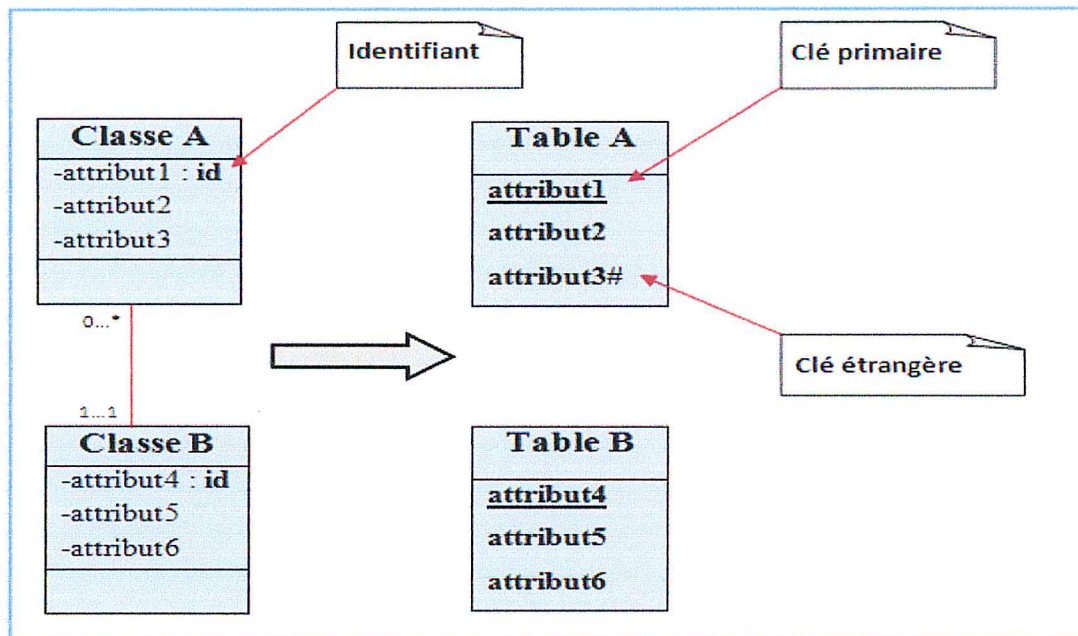


Figure VII.2 : Transformation d'une relation un-à-plusieurs

3.1.2.2. Relation plusieurs-à-plusieurs (règle 3):

On applique la première règle de passage. En plus, la relation entre les deux classes devient une table et les clés primaire des deux classes migrent vers cette table et forment ensemble la clé primaire.

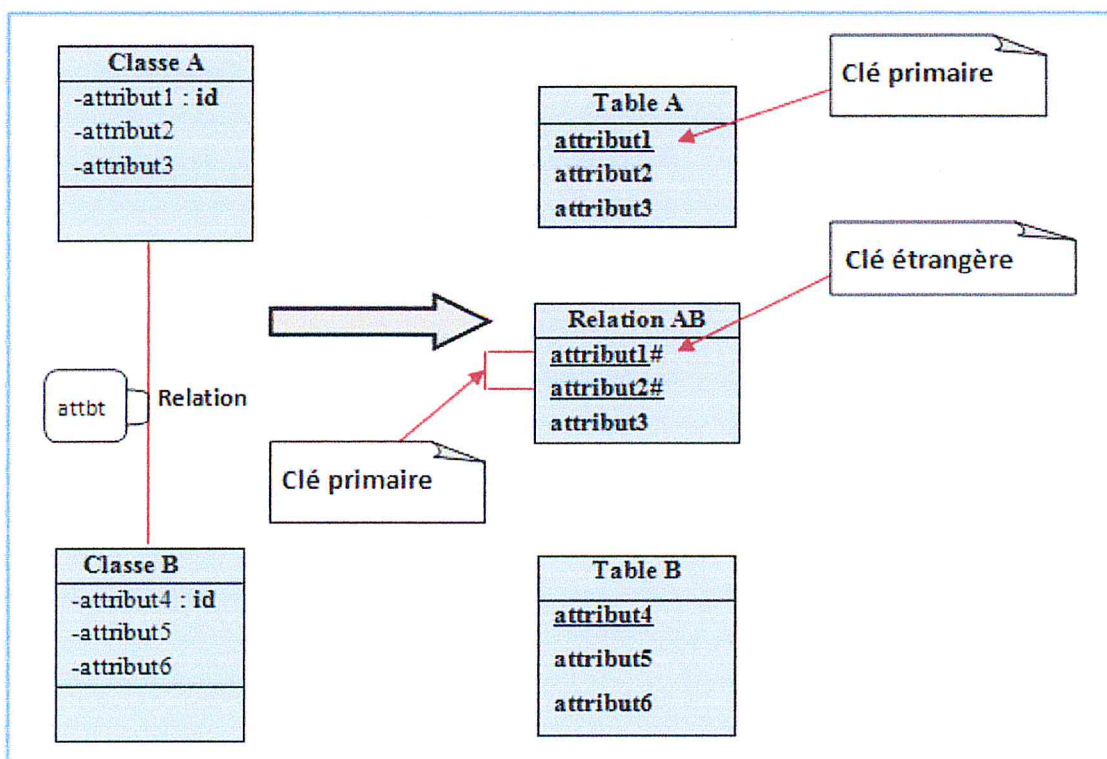


Figure VII.3 : Transformation d'une relation plusieurs-à-plusieurs

3.1.2.3. Relation un-à-un (règle 4):

Dans le cas d'une relation un-à-un, on applique la première règle. En plus, la clé de la classe A migre vers la classe B comme clé étrangère.

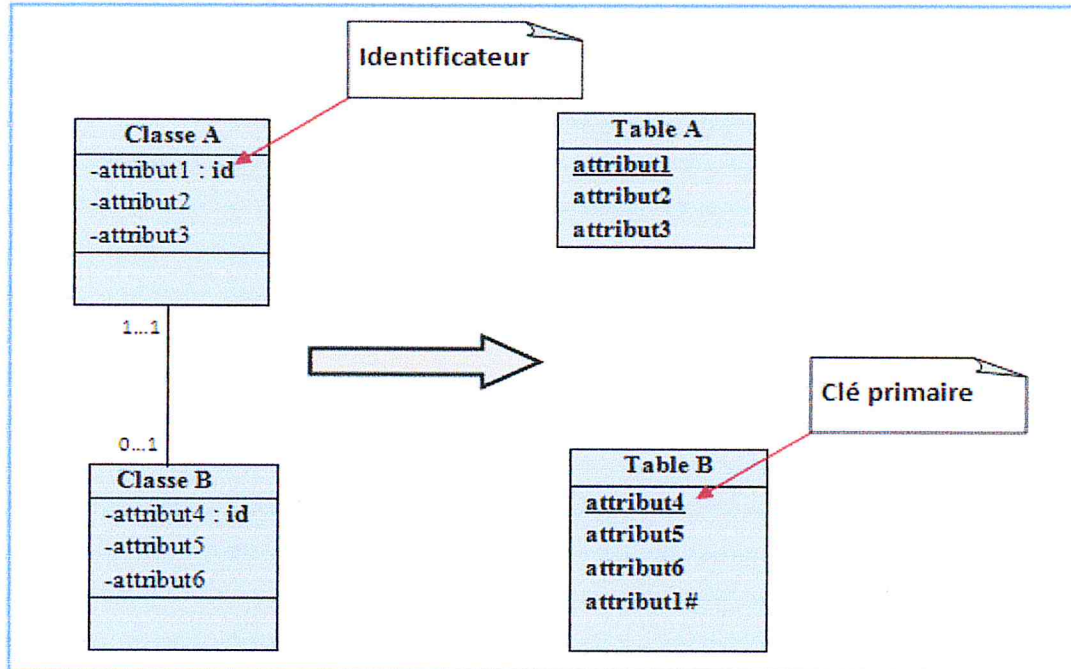


Figure VII.4 : Transformation d'une relation un-à-un.

3.1.2.4. Relation d'héritage (règle 5):

La relation d'héritage peut être transformée de plusieurs manières. A ce niveau nous décrivons le push-up. Car nous l'avons appliqué pour transformer les relations d'héritage de notre diagramme de classe.

Le principe consiste à créer une seule classe qui contient les attributs de la super classe et ceux de toutes les sous-classes, en ajoutant un attribut pour distinguer les différentes sous-classes. Puis, on applique la règle une.

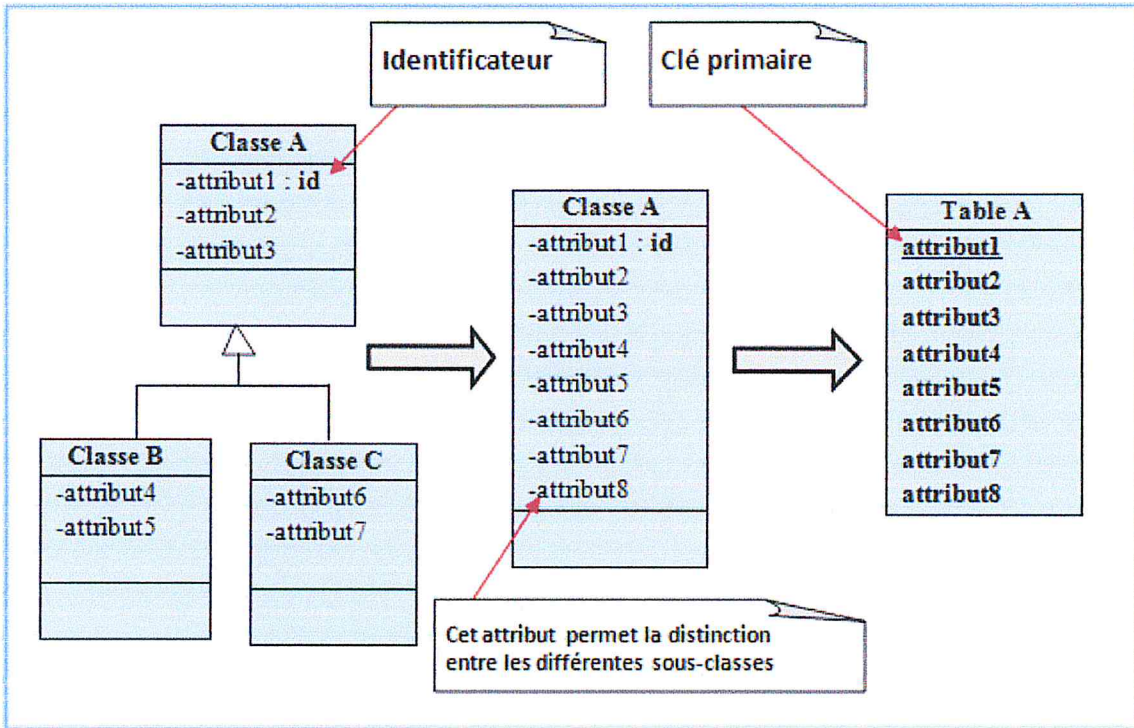
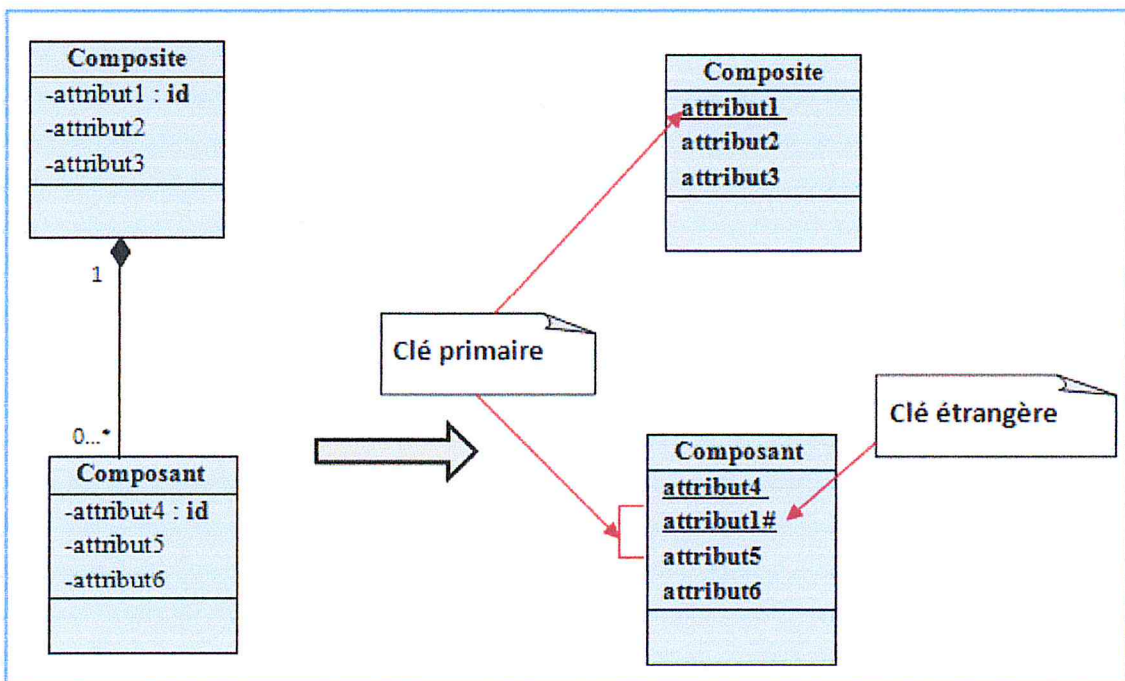


Figure VII.5: Transformation d'une relation d'héritage.

3.1.2.5. Relation de composition (règle 6):

Dans le cas d'une relation de composition, on applique la règle une sur le composite et le composant. En plus, la clé du composite migre vers le table du composant et forme une clé primaire avec la clé de ce dernier.



FigureVII.6 : Transformation d'une relation de décomposition.

3.1.3. Schéma relationnel de la base de données :

Le schéma relationnel de la base de données est présenté ci-dessous

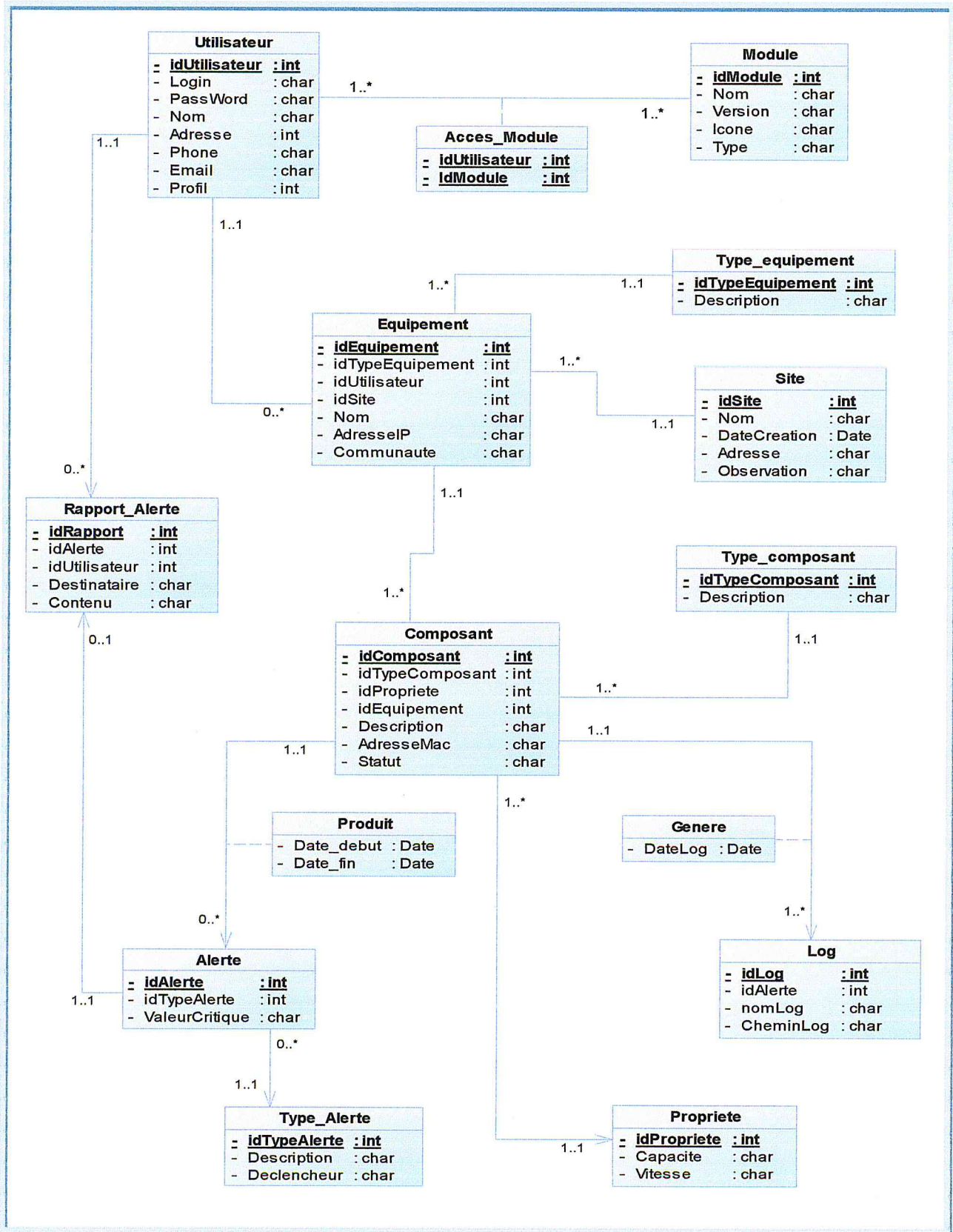


Figure VII.7 : Schéma relationnel de la base de données

4. Réalisation :

Dans cette étape nous présenterons les IHM les plus importantes de notre plate forme.

4.1. Page d'accueil :

Fenêtre de connexion à l'application. Cette fenêtre s'affiche au démarrage de l'application

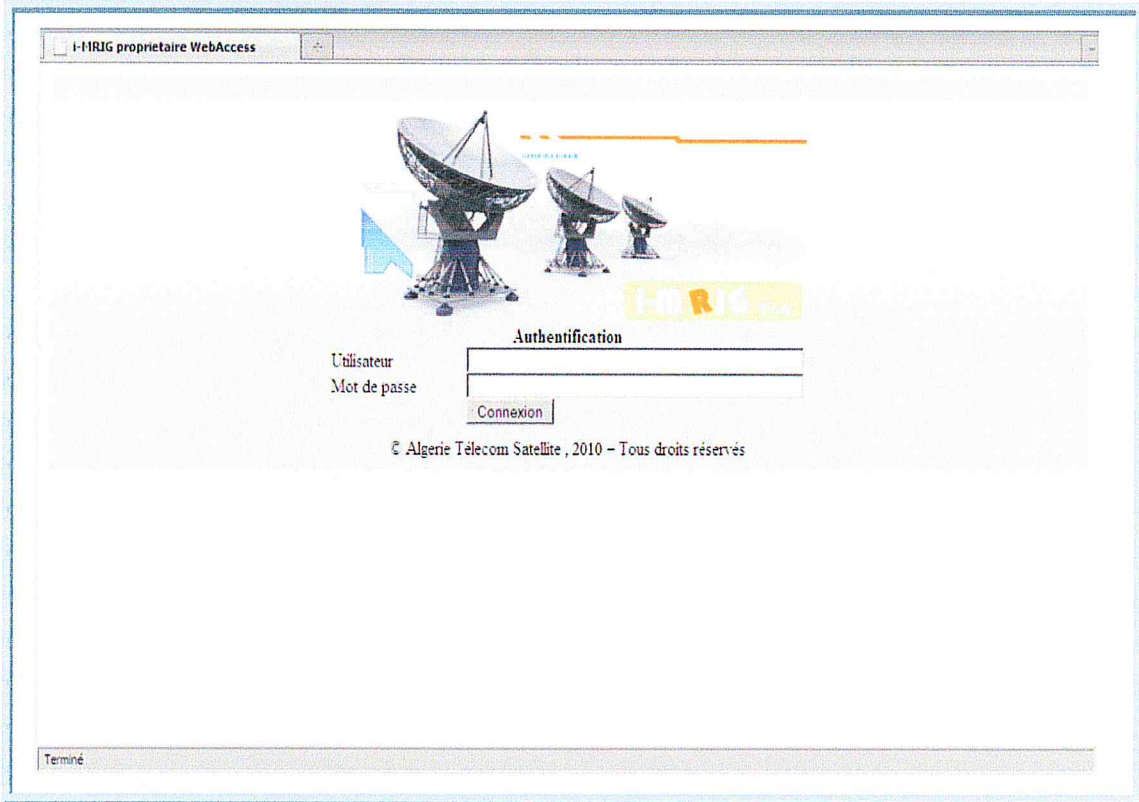


Figure VII.8 : Fenêtre de page d'accueil.

4.2. Liste des clients, équipements et composants:

L'arborescence à gauche affiche la liste des clients

- ✓ Un clic sur le nom du client affiche tout ses équipements
- ✓ Un clic sur un équipement affiche tous ses composants

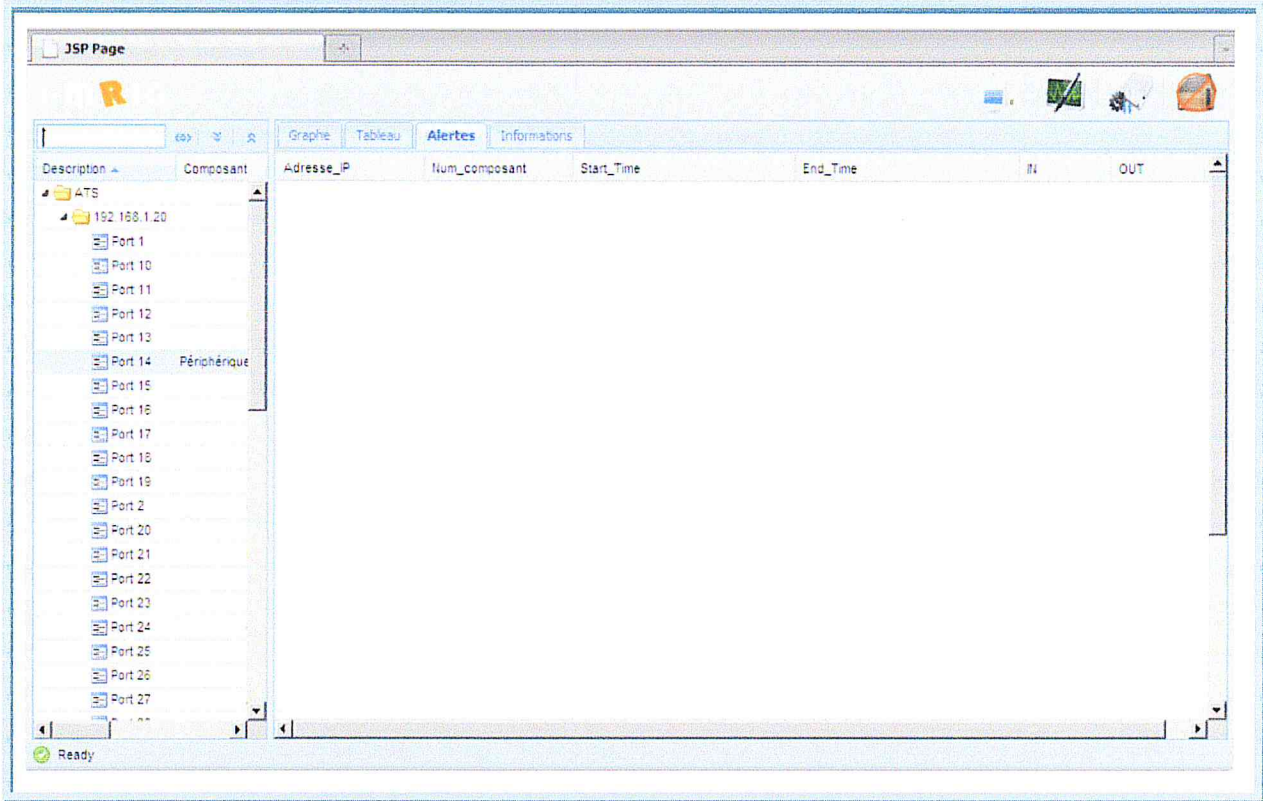


Figure VII.9 : Fenêtre de la liste des clients, équipements et leurs composants

4.3. Liste des alertes:

- L'utilisateur peut afficher la liste des alertes d'un composant donné

Adresse_IP	Num_composant	Start_Time	End_Time	IN	OUT	
10.1.106.3	1	25/09/2010 04:37:04	25/09/2010 04:42:04	0	0	!
10.1.106.3	1	25/09/2010 04:37:01	25/09/2010 04:38:01	0	0	!
10.1.106.3	1	25/09/2010 01:18:19	25/09/2010 01:19:19	0	0	!
10.1.106.3	1	25/09/2010 01:17:18	25/09/2010 01:18:18	0	0	!
10.1.106.3	1	24/09/2010 20:59:14	24/09/2010 21:25:14	0	0	!
10.1.106.3	1	24/09/2010 20:26:29	24/09/2010 20:28:29	0	0	!
10.1.106.3	1	24/09/2010 19:07:51	24/09/2010 19:39:51	0	0	!
10.1.106.3	1	24/09/2010 19:07:49	24/09/2010 19:08:49	0	0	!
10.1.106.3	1	21/09/2010 12:23:40	21/09/2010 12:40:40	0	0	!
10.1.106.3	1	21/09/2010 12:17:39	21/09/2010 12:23:39	0	0	!
10.1.106.3	1	21/09/2010 12:17:33	21/09/2010 12:18:33	0	0	!
10.1.106.3	1	15/09/2010 16:01:16	15/09/2010 16:20:16	0	0	!
10.1.106.3	1	15/09/2010 15:32:08	15/09/2010 15:49:08	0	0	!
10.1.106.3	1	15/09/2010 15:08:07	15/09/2010 15:32:07	0	0	!
10.1.106.3	1	15/09/2010 15:08:03	15/09/2010 15:09:03	0	0	!
10.1.106.3	1	14/09/2010 00:27:36	14/09/2010 00:43:36	0	0	!
10.1.106.3	1	14/09/2010 00:27:34	14/09/2010 00:28:34	0	0	!

Figure VII.10 : Fenêtre de la liste des alertes d'un composant

- L'utilisateur a la possibilité d'afficher les alertes de tous les composants d'un équipement donné, comme il peut trier les alertes dans un ordre croissant ou décroissant selon l'une des propriétés de l'alerte

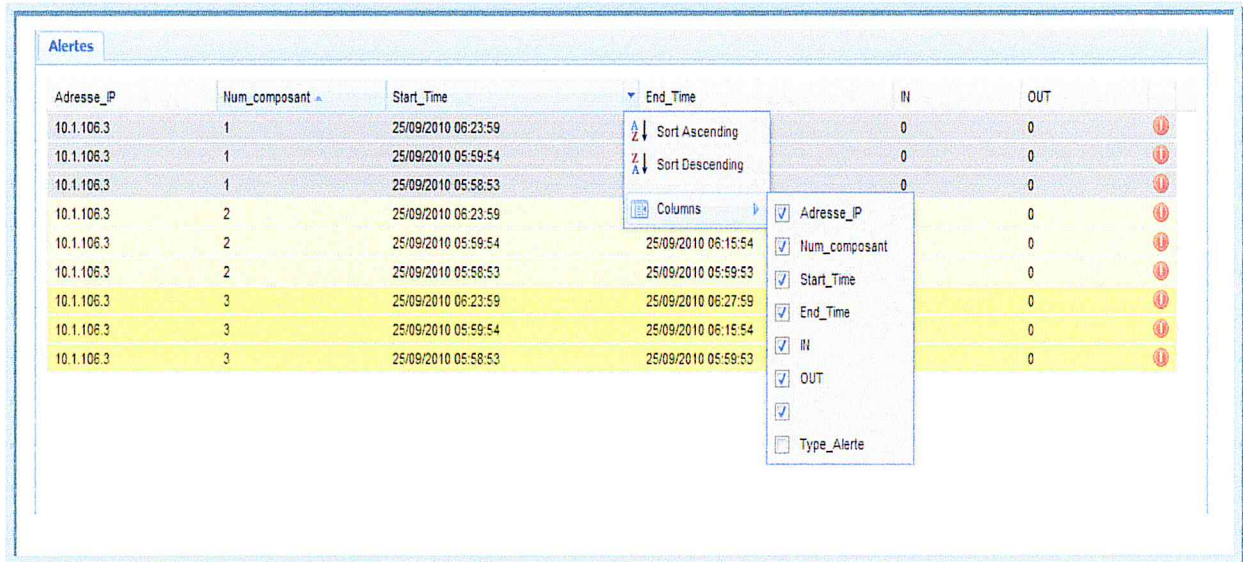


Figure VII.9 : Fenêtre de la liste des alertes d'un équipement

4.4. Détails d'une alerte:

Fenêtre qui s'affiche lorsqu'un utilisateur souhaite avoir des informations supplémentaires sur une alerte donnée

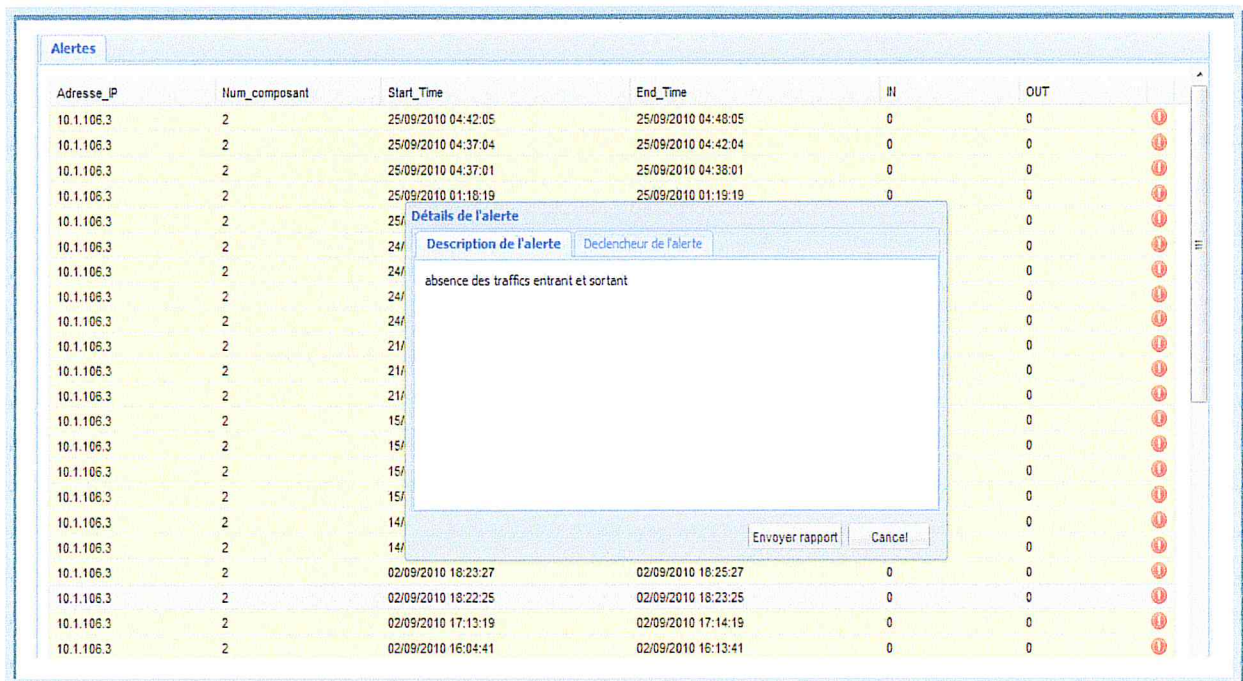


Figure VII.11 : Fenêtre des détails d'une alerte

4.5. Détails d'un composant:

Le système permet à l'utilisateur d'avoir plus de détail concernant un composant qui à été à l'origine d'une alerte donnée

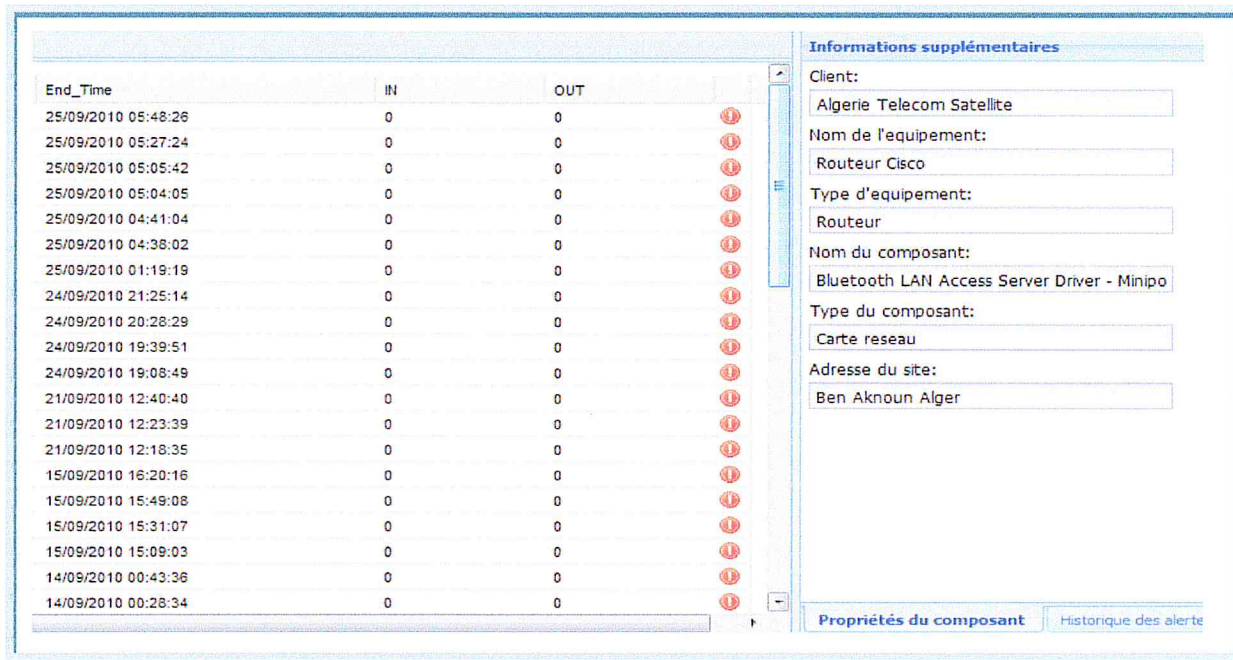


Figure VII.12 : Fenêtre des détails d'un composant

4.6. Chargement de l'historique des alertes:

- Le superviseur sélectionne un composant de la liste des composants
- Il saisie les dates début et fin puis clic sur le bouton charger l'historique
- Le tableau des alertes n'affiche que les éventuelles alertes du composant produites entre les deux dates saisies

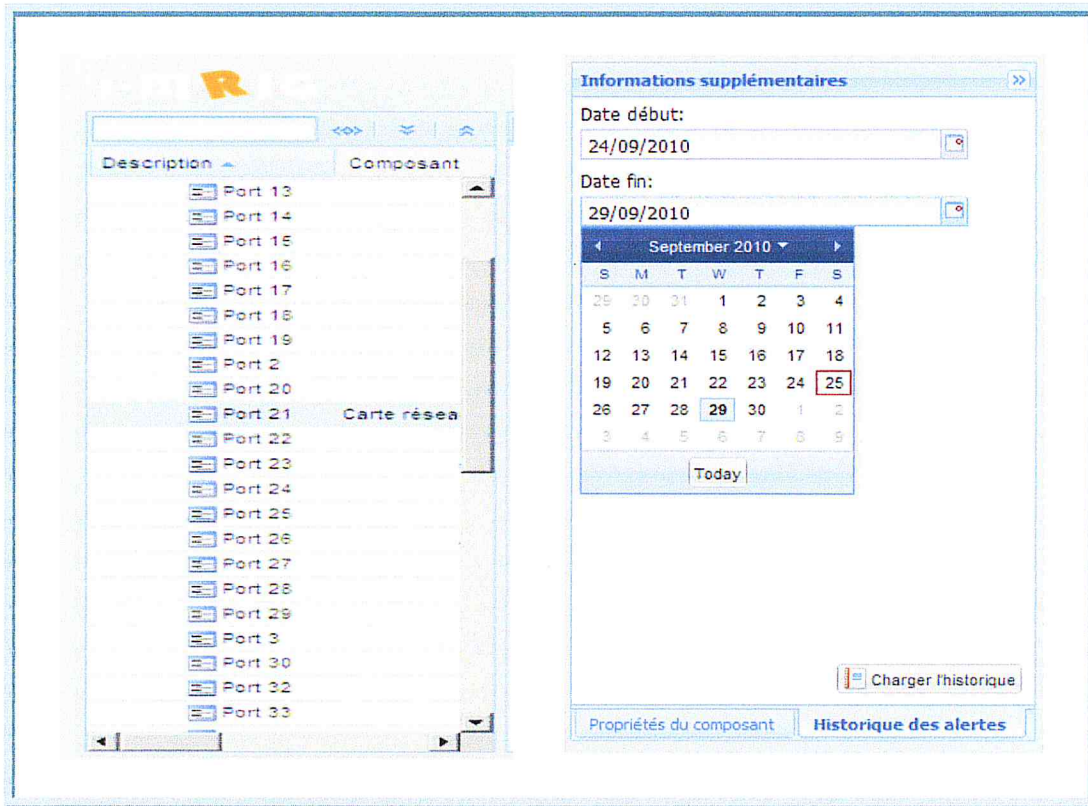


Figure VII.13 : Fenêtre de chargement de l'historique des alertes

4.7. Envoi d'un rapport d'alerte:

Le système permet au superviseur d'envoyer un rapport d'alerte concernant une alerte donnée aux services chargés du traitement des alertes, s'il le juge utile

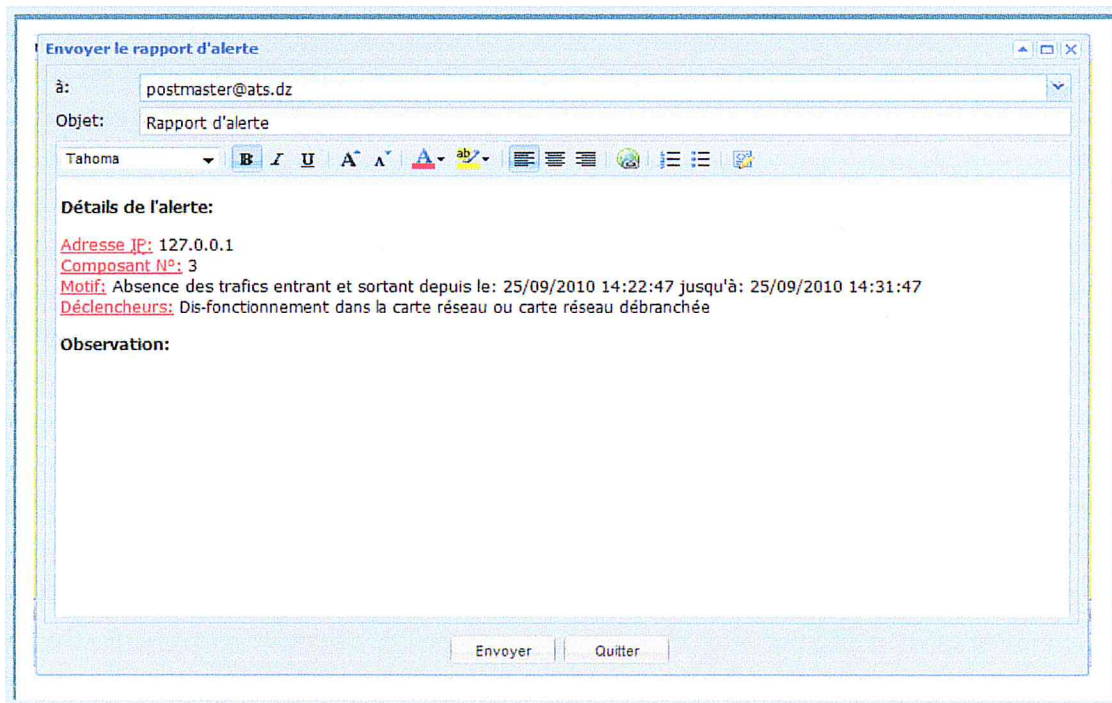


Figure VII.14 : Fenêtre de l'envoi d'un rapport d'alerte

Conclusion Générale

La gestion de réseaux est l'un des domaines les plus complexes auxquels l'on puisse se confronter; elle cumule la distribution, le modèle objet, le temps réel, le transactionnel, la gestion d'équipements complexes. C'est en conséquence une source de coût importante pour l'opérateur, qui se voit contraint d'investir des sommes significatives et des compétences critiques dans une fonction qui semble non immédiatement rentable.

Néanmoins, le souhait exprimé ici concerne la compréhension de la nécessité de la fonction Gestion de Réseaux, et son intérêt pour l'entreprise; ce n'est qu'à travers de la gestion efficace que cette dernière sera en mesure d'offrir les services qui le différencieront de la concurrence.

L'objectif de ce projet était essentiellement de développer une application web pour la gestion des alertes relatives aux équipements *VSAT* des clients d'Algérie Télécom Satellite, pour se faire il était nécessaire de faire une étude bibliographique approfondie. Au cours de notre étude nous avons essayé de nous arrêter sur les deux points importants de notre travail, à savoir la technologie *VSAT* et la supervision des réseaux via le protocole *SNMP*.

Pour réaliser ce travail, nous avons adopté un cycle de développement en *Y* qui est *2TUP*. En commençant avec une Etude Préliminaire qui avait pour but, de bien comprendre le contexte du projet et d'en tirer les besoins essentiels pour pouvoir entamer sa modélisation. En second une étude technique et une étude fonctionnelle ont été traitées en parallèle pour cerner et modéliser les besoins fonctionnels et techniques du système.

Nous avons fini avec une conception détaillé qui nous a permis de mettre en œuvre la solution. Tout au long du développement, la modélisation du système a été effectuée à l'aide du langage UML.

La mise en œuvre a été implémenté avec le langage JSP et l'utilisation d'un SGBD Oracle pour le stockage des données.

Le système a été mis en œuvre entièrement selon le cahier des charges établi dans cette étude.

Ce projet nous a permis de mettre en pratique les connaissances acquises tout au long de notre cursus universitaire. Mais aussi de raffiner notre méthodologie de travail et de

développer notre esprit d'équipe en travaillant sur un projet qui demande des compétences variées.

En perspectives il y'a lieu de :

- Enrichir la solution en proposant :
 - ✓ une cartographie de réseaux,
 - ✓ gestion poussée d'alarmes,
 - ✓ gestion des utilisateurs,
 - ✓ gestion de pannes,
 - ✓ statistiques et reporting
- Possibilité de choisir un moteur de récolte des informations de gestion depuis les équipements.

Bibliographie

[Eeles 02]: P. Eeles, K.Houston, W.Kozaczynski, “*Building J2EE Applications with the Rational Unified Process*”, 2002, Addison-Wesley.

[LAF10] : LAFOSSE.J, «Coupler la puissance de Java EE et PHP grâce à GlassFish», mai 2010.

[PER 1997]: D.PERKINS, E.MCGINNIS, « Understanding SNMP MIBs », 1997.

[ROQ2003] : P. ROQUES et F. VALLEE, « UML2 en action : De l'analyse des besoins à la conception en Java », 2ème édition, EYROLLE, 2003.

[ROQ2005] : P. ROQUES, « UML 2 par la pratique,EYROLLE », 2005.

[ROQ2008] : P. ROQUES, « Les cahiers du programmeur : UML2 modéliser une application web », 4ème édition, EYROLLE, 2008.

[SIM 1997]: N.SIMONI, S.ZNATY, « Gestion de réseau et de service similitude des concepts, spécificité des solutions », InterEditions Masson, 1997.

Webographie

- [Web1]: <http://www-igm.univ-mlv.fr/~duris/NTREZO/20022003/VSAT.pdf>
- [Web 2]: http://www.cnam.fr/elau/publi/autres/images/Communication_satellite.pdf
- [Web 3]: <http://www-sop.inria.fr/members/Eitan.Altman/DEASAT/deasat2.pdf>
- [Web 4]: <http://dvb-scpc.atrexx.com/>
- [Web 5] :
http://www.etudionet.com/communaute/xuser/etudionet/docs/PFE_BEN_AMARA_Sahbi.pdf
- [Web 6]: <http://www-igm.univ-mlv.fr/~dr/XPOSE2002/vollerin/solGestion.html>
- [Web 7]: <http://www.o00o.org/monitoring/solutions.html>
- [Web 8]: <http://greenstone.refer.bf/collect/upb/index/assoc/HASH866f.dir/ESI-2007-FAR-GES.pdf>
- [Web 9]: <http://www.freewebs.com/fresma/Formation/UML/Processus%20Unifie.pdf>
- [Web 10]: <ftp://ftp-developpez.com/bassim/Ressources/Memoire/2TUP.pdf>
- [Web 11] : <http://doc.ubuntu-fr.org/mrtg>
- [Web 12]: <http://www.techno-science.net/?onglet=glossaire&definition=5266>
- [Web 13]:
<http://www.memoireonline.com/07/09/2392/Mise-en-place-d'une-architecture-3-tiers-avec-base-de-donnees-centralisee-sous-SQL-SERVER-Cas-du.html>
- [Web 14]: <http://www.wikipédia.mht>
- [Web 15]: <http://www.commentcamarche.net/contents/jsp/jspintro.php3>
- [Web 16] : <http://www.commentcamarche.net/contents/javascript/jsintro.php3>
- [Web 17]: <http://www.commentcamarche.net/contents/css/cssintro.php3>
- [ORA09] : <http://www.oracle.com/>
- [Web 18]: <http://www.commentcamarche.net/contents/oracle/oracintro.php3>
- [Web 19]: <http://www.projet-plume.org/fiche/netbeans>
- [Web 20]: <http://www.farkess.com/tutoriels/558/utilise-extjs-exemples-pas-a-pas/>
- [Web 21]: http://www.oracle.com/lang/fr/database/Standard_Edition.html
- [Web 22]: <http://www.apl.jhu.edu/~hall/java/Servlet-Tutorial/>