INSTITUT NATIONAL DES SCIENCES APPLIQUEES DE LYON

DEPARTEMENT INFORMATIQUE

TELEINFORMATIQUE

Tome 4 (extraits)

ADMINISTRATION DES RESEAUX

2000-2001

G. Beuchot

Administration de réseaux

1. Principes et concepts

1.1 Définition

L'administration de réseaux englobe les moyens mis en oeuvre pour :

- offrir aux utilisateurs une qualité de service donnée et garantir cette qualité de service.
- permettre et guider l'évolution du système en fonction
  * du trafic
  * des nouvelles applications
  * des nouvelles technologies
- représente le partie opérationnelle d'un système, soit
  * la surveillance du réseau informatique :
    systèmes informatiques et réseaux d'interconnexion
  * le support technique
  * la gestion des coûts, des ressources, etc
  * la gestion de ressources humaines

L'administration de réseau est appliquée en suivant une politique, c'est à dire des objectifs à atteindre ("activité administration de réseaux").

Cette politique spécifie des actions à long, moyen et court terme par :

* une stratégie, plan des actions à entreprendre à long terme, de quelques mois à un ou deux ans

*une tactique, plan d'exécution pour atteindre les objectifs à moyen terme, de quelques jours à un ou deux mois

* un fonctionnement opérationnel, pour gérer le réseau en continu, à court terme, de quelques minutes à quelques heures.

Ceci implique la définition de modes opératoires et leur mise en oeuvre.

1.2 Disciplines

L'administration de réseau ne porte pas seulement sur le réseau de télécommunications au sens strict, mais englobe aussi l'administration

- des utilisateurs
  - qui
  - où
  - comment les atteindre
  - comment les identifier
  - quels sont leurs droits

- des serveurs et des ressources
  - quelles machines
  - quelles ressources
  - quelles fonctions de communication, comment les utiliser
  - quelle sécurité sur les données et les ressources
  - quels sont leurs coûts d'utilisation.

- du (ou des) réseau(x) de télécommunication

C'est une ressource particulière ayant des composants variés:

- Informatiques de télécommunication pour réseaux informatiques, modems, concentrateurs, multiplexeurs, commutateurs,etc
- de télécommunication généraux : PABX, accès aux réseaux publics

1.3 Fonctionnalités

Elles sont regroupées en cinq grandes classes

Gestion des anomalies
Gestion de la comptabilité
Gestion de la sécurité
Gestion des performances
Gestion de la configuration et des noms

1.3.1 Gestion des anomalies

Elle recouvre la détection des anomalies, l'identification et la correction de fonctionnements anormaux. Ces défauts font qu'un système n'atteint pas ses objectifs; ils sont temporaires ou permanents. Ils se manifestent comme des événements.

Elle fournit une assistance pour répondre aux besoins de la qualité de service et à sa permanence.

La gestion d'anomalies comprend les fonctions suivantes :

- réception de et actions sur des notifications de détection d'erreurs
- recherche et identification des anomalies
- exécution des séquences de tests de diagnostic
- correction des anomalies
- tenue et examen des journaux d'erreurs

1.3.2 Gestion de la comptabilité

C'est une activité qui peut être complexe car elle doit prendre en compte la totalité du réseau informatique, de ses services et de ses ressources.

Elle comprend les fonctions suivantes :
- information des utilisateurs sur les coûts encourus ou les ressources utilisées
- possibilité de fixer des limites comptables et des prévisions de tarifs, associées à l'utilisation des ressources
- possibilité de combiner les coûts quand plusieurs ressources sont utilisées pour atteindre un objectif de communication donné.

Ceci conduit à la mise en place de classes d'utilisateurs avec des facturations à la consommation ou forfaitaires avec surcoûts pour les dépassements de consommation (temps de communication, temps de traitements, occupation mémoire ou disques, volume des informations transférées, etc.)

1.3.3 Gestion de la sécurité (sûreté)

Elle doit répondre à deux types de problèmes :

- garantir les abonnés (utilisateurs)
  les services et les ressources

- garantir le réseau lui même

contre les intrusions volontaires, agressives ou passives, mais aussi contre des actions involontaires mais dangereuses d'utilisateurs habilités.

Pour cela elle comporte les fonctions suivantes :

- création, suppression et contrôle des mécanismes et services de sécurité (identification, authentification, clés d'accès, groupes fermés d'abonnés, cryptage,...)

- diffusion des informations relatives à la sécurité

- compte rendu d'événements relatifs à la sécurité (audit)

La mise en oeuvre des fonctions de sécurité ne fait pas à proprement parlé de la gestion de la sécurité.

1.3.4 Gestion des performances

Cette activité sert de base à la fourniture d'une qualité de service garantie. Pour cela elle traite des problèmes à moyen et à long terme. Elle analyse le trafic, le fonctionnement du réseau (débits, temps de réponses) et utilise ces informations pour régler le système en déterminant de nouvelles procédures d'acheminement par exemple et en les mettant en place ou, à long terme, en planifiant l'évolution du réseau (topologie, capacités des canaux).

Elle met en oeuvre les fonctions suivantes :

- collecte des statistiques
- définition de la performance du systèmes dans des conditions naturelles ou artificielles (mode dégradé)
- modification des modes de fonctionnement du système pour mener des activités de gestion de performances (acheminement par exemple).

Pour traiter ces fonctions elle doit traiter les données statistiques, modéliser le système et simuler son comportement.

1.3.5 Gestion de la configuration et des noms

Elle est à la base des quatre autres activités; elle leur permet de connaître tous les composants des systèmes et de les gérer. Elle permet de les désigner par leur adresses physique ou leu nom (adresse logique) et de maintenir la cohérence de ces noms . Ceux-ci sont consignés dans différents fichiers répartis dans les systèmes interconnectés et leur mises à jour doit en garder la cohérence; on utilise aussi des serveurs de noms, primaires et secondaires dont il faut aussi maintenir la cohérence.

Elle comporte les fonctions suivantes :

- établissement des paramètres contrôlant le fonctionnement normal du système

- association de noms aux objets de gestion ou à des ensembles d'objets de gestion

- initialisation et retrait d'objets de gestion

- récolte d'information sur l'état du système, périodiquement ou à la demande

- acquisition des notifications des modifications importantes de l'état du système

- modification de la configuration du système.

Par ces fonctions, elle permet de préparer, d'initialiser, de démarrer et de terminer les services d'interconnexion et d'en assurer la continuité de fonctionnement.

1.4 PORTEE et RESPONSABILITES

La portée des actions à traiter peut être divisée en quatre niveaux :

- planification
- analyse des performances
- gestion des problèmes et des ressources
- contrôle opérationnel

A chacun de ces niveaux sont associés des ressources humaines ayant des responsabilités propres.

1.4.1 Contrôle opérationnel

Il est chargé de la surveillance et du support technique du réseau; il est assuré par trois groupes de personnels.

* Bureau d'aide

Il réalise l'interface avec les utilisateurs pour les conseiller et traiter les anomalies de type 1, par exemple les mises sous tension de certains composants, les problèmes de modems, les initialisations des coupleurs de communication (débit, parité, etc). Ces anomalies de types 1 représentent environ 80% des problèmes rencontrés.

* Opérateur réseau

Il s'occupe de la surveillance et de la commande du réseau. Il observe la charge du réseau, les anomalies, les incidents, les reprises automatiques etc. et collabore avec le support technique.

* Support technique

Il traite les problèmes techniques liés à l'installation de nouveaux équipements ou à leur maintenance . Il fournit cette aide à la maintenance sur la demande du bureau d'aide ou des opérateurs. Partageant cette maintenance avec le fournisseur du matériel ou du logiciel, il analyse les anomalies et assiste le vendeur en cas de télémaintenance en t sur le site. (Il est souvent difficile de savoir où est le cause réelle d'un anomalie observée).

Il traite les anomalies de type 2 qui représentent 10 à 15% des cas (en liaison avec les opérateurs).

1.4.2 Gestion des services, des ressources et des problèmes

L'équipe chargée de ces problèmes assure le contact avec les fournisseurs.

Elle réalise :

l'inventaire des ressources
l'ordonnancement des changements de topologie (niveau tactique)
la mise en oeuvre des modifications d'acheminement
la gestion des services : sont-ils possibles, faut-il les ajouter ou en supprimer d'autres ?

La gestion des problèmes complexes (anomalies de type 3). Ceux-ci comportent les bogues logiciels, les différences de comportement entre systèmes, etc. Ces problèmes sont très vites résolus lorsqu'ils sont bien identifiés (pa exemple erreur ou omission dans la documentation ...) ou alors très long à corriger.

1.4.3 Analyse des performances

Cette responsabilité travaille à moyen et à long terme.

Elle s'occupe

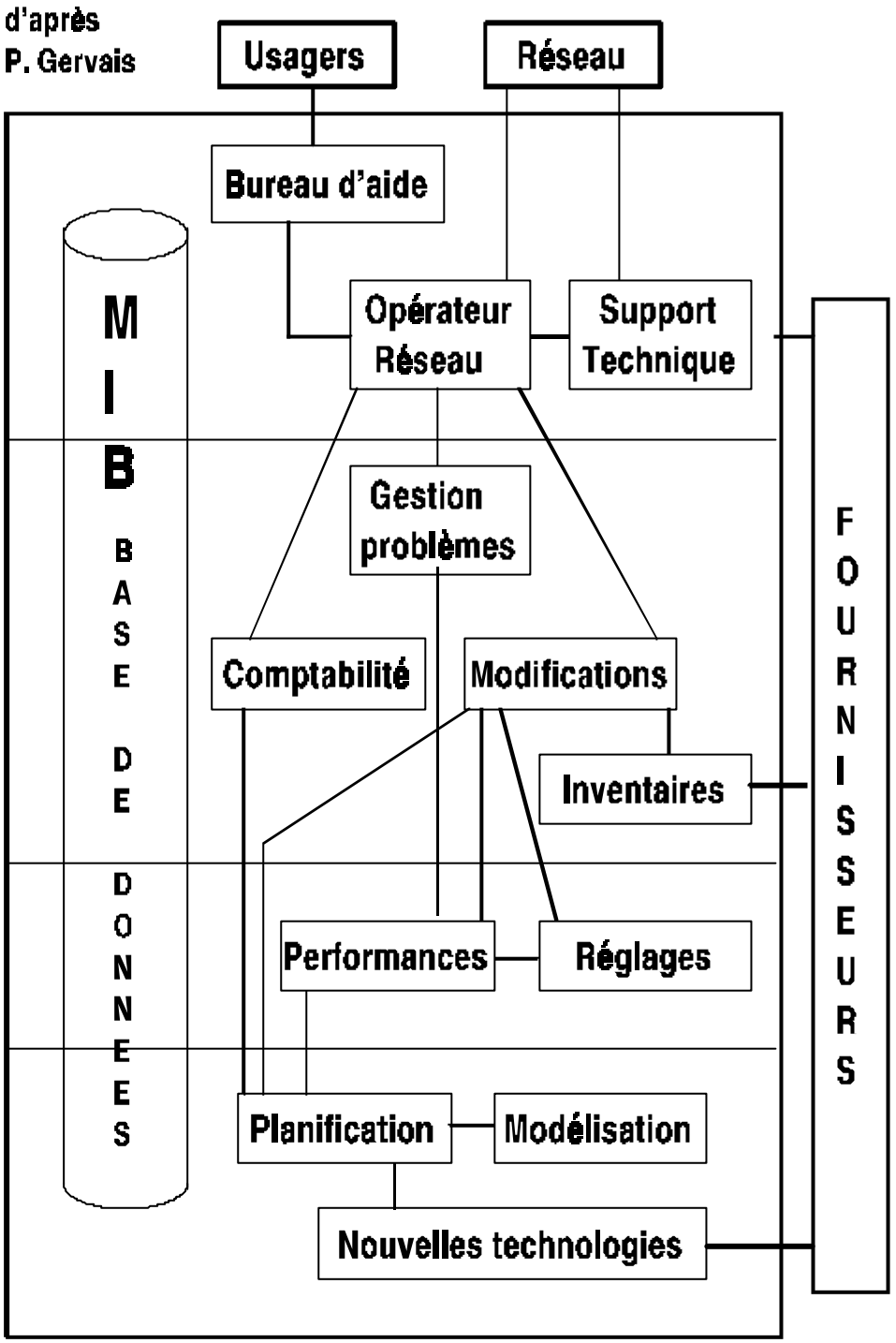
- de la mesure des performances

définition des objectifs à atteindre : temps de réponse, nombre de transactions journalières ou horaires, etc;
paramètres à prendre en compte, statistiques. Ces paramètres sont pris dans la base de données administratives. Il faut donc qu'elle demande qu'ils y soient consignés.

-des mesures à prendre pour améliorer ces performances
en particulier de l'élaboration des tables d'acheminement et des mesures à prendre pour les mettre en oeuvre de manière coordonnée.

1.4.4 Planification

Elle traite des évolutions à long terme du réseau pour tenir compte des augmentations de trafic, soit pour la mise en oeuvre de nouvelles techniques ou de nouveaux services par exemple le RNIS ou la messagerie X400...



Ces nouvelles techniques sont elles applicables à l'entreprise, sont elles disponibles, dans quels délais, à quels coûts, qu'elle est l'évolution prévisible de ces coûts ?

Ceci entraîne en général une modèlisation du réseau et des études de comportement par simulation (ou calcul direct approchés)

Les responsables de la planification doivent être à l'écoute des tendances techniques et des besoins exprimés des utilisateurs ou baux de l'entreprise.

1.5 Complexité du problème

L'administration de réseau est une activité très complexe car elle doit répondre globalement aux besoins d'un système distribué et hétérogène.

Cette hétérogénéité est à prendre en compte non seulement au niveau des communications (standards, équipements), mais aussi au niveau des systèmes et des utilisateurs (besoins divergents).

L'ADMINISTRATION DE RESEAU EST GLOBALE

Pour répondre à cette hétérogénéité et à la distribution du système le réseau est partitionné en domaines

- géographiques
- applicatifs
- selon les constructeurs (SNA, DSA, DECNET,etc)
- par normes (OSI, TCP/IP, SNA)

Il n'y a pas de régles générales pour décider de cette partition et celle-ci est souvent double (géographique/norme par exemple).

Cette partition en domaine entraîne un problème de localisation des responsabilités en particulier entre domaine public et domaines privés . Le réseau public apparaît souvent comme une tache blanche dans les domaines administrés qui sépare les domaines privés.

Dans un domaine qui administre ? Qu'elle est son autorité par rapport à l'administrateur central mais aussi par rapport aux responsables des systèmes informatiques ? Comment peut-il imposer les changements ? En général l'administrateur de réseau doit avoir une responsabilité supérieure au responsable système car il doit tenir compte de toutes les contraintes des systèmes interconnectés.

D'autre part l'administration de réseau doit fonctionner même en cas d'anomalie grave (panne) du réseau.

Pour cela celui-ci peut être maillé ou l'administration doit être supportée par un réseau d'administration distinct.

En cas de réseau d'administration disjoint, qui l'administre, qu'elles sont ses relations avec le réseau administré ? Quel est sont coût (en général réseau public commuté comme réseau de télécommunication support)?

En cas de réseau maillé, qu'elle doit être sa configuration ? Quels sont les acheminements à prévoir en secours en cas de panne?

1.6 Partition en domaines

1.6.1 Types de systèmes et composants

Un système appartient à un ou plusieurs domaines. Il peut être système gestionnaire ou système géré (ou administré).

Un domaine doit comporter un système gestionnaire et un ou plusieurs systèmes gérés.

Un système peut être géré dans un domaine et gestionnaire dans un autre. On crée ainsi une relation hiérarchique entre domaines. Le découpage en domaines peut être complexe (par exemple domaines géographique et norme ou constructeur); il peut donc y avoir des "croisements", un système étant gestionnaire dans un domaine et géré dans l'autre et vice versa.

Dans un système gestionnaire on trouve :

- une base de donnée administrative (MIB : management information base) qui contient les objets du système et leurs attributs

- des processus qui permettent l'exécution des fonctions de base (extraction des informations, rangement dans la MIB directement ou après prétraitement) ou qui supportent les outils logiciels de traitement de ces données

- un langage pour les interactions avec les opérateurs

Dans un système géré on ne trouve que :

- une base de données administrative (MIB)
- les fonctions de base

Les traitements sont faits dans le système gestionnaire après transfert depuis le système géré. On doit donc disposer aussi d'un protocole de communication

1.6.2 Processus gestionnaire et processus agent

Dans un système géré on dispose d'un processus agent chargé de traiter les objets administrés et leurs attributs.

Dans un système gestionnaire on dispose d'un processus gestionnaire chargé de l'administration de domaine et d'un processus agent chargé des objets locaux. Le processus gestionnaire communique avec les processus agents distants ou l'agent local.

Un domaine existe s'il a au moins un processus gestionnaire et un ou plusieurs processus agents.

Les objets d'un système ne peuvent être manipulés que par l'intermédiaire d'un processus agent. Ils peuvent raccordés à plusieurs de ces processus.

1.6.3 Réseau d'administration et réseau administré

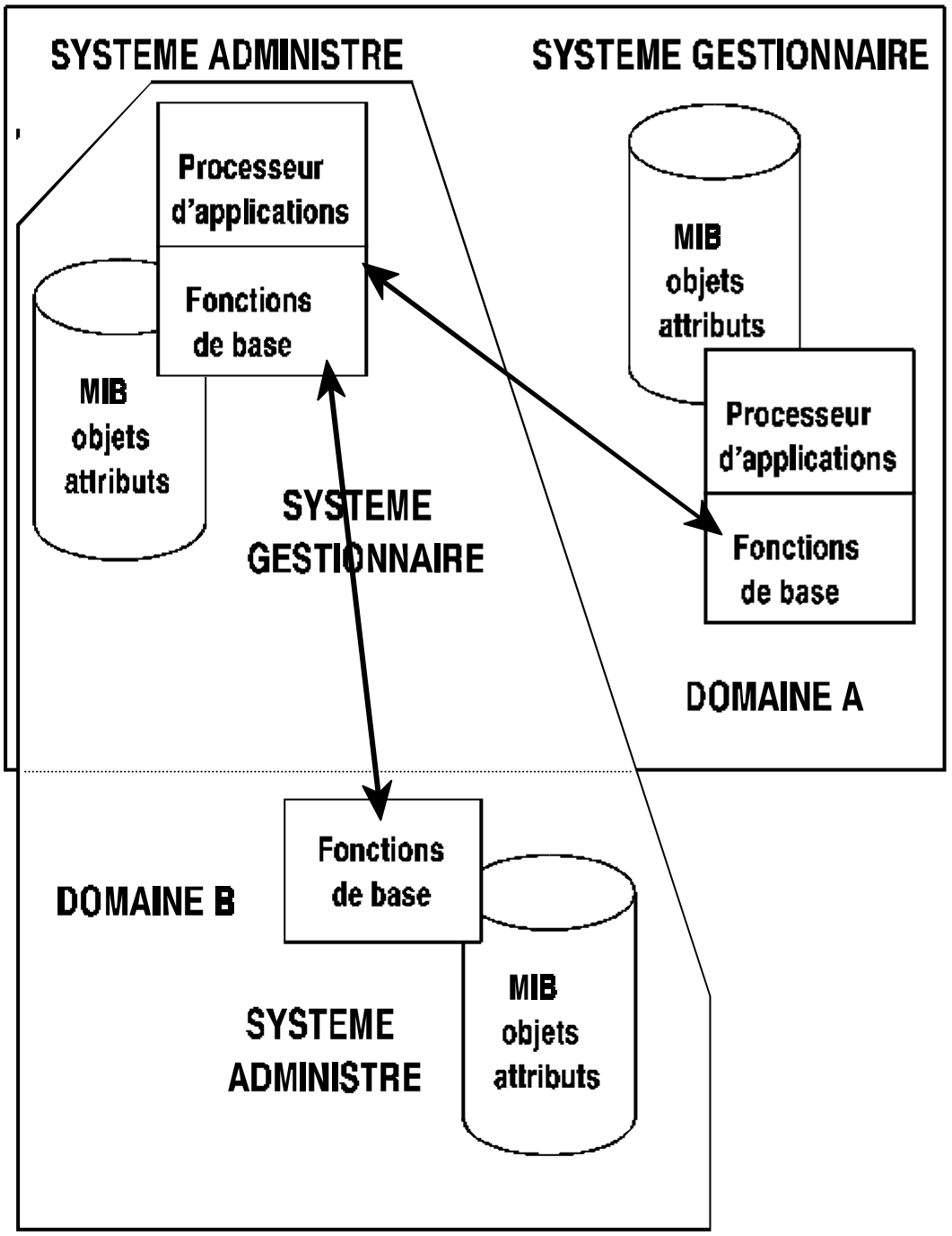
Depuis un système central, système gestionnaire du domaine principal, on peut atteindre

- les noeuds de ce domaine
- les racines des sous-domaines

Ces racines rapatrient et condensent les données adminis- tratives de ces sous-domaines

Depuis un noeud (système géré avec son processus agent) on traite des objets. Ces objets sont le passage obligé pour atteindre les informations administratives, codées dans les attributs de ces objets. Les objets représentent le réseau administré et l'ensemble des processus d'administration et des protocoles qui les relient constituent le réseau d'administration.

Ainsi toutes les informations d'administration peuvent remonter à un point focal d'où est géré l'ensemble des domaines.



Cette architecture est entièrement orientée objet. Le modèle OSI d'administration de réseau fournit les bases et les fonctions pour la réaliser.

2. administration OSI

Nota : Les travaux de l'OSI dans le domaine de l'administration de réseaux ont été menés dans deux directions :

Etudes des principes de base et définition d'une architecture, de services et de protocoles.

Etude et définitions précises des objets administrés et de leurs attributs. Cette seconde partie a été réalisée par un ensemble de constructeurs regroupés dans l'OSI-NMF : Network Management Forum. Les premiers documents de NMF ne sont disponibles que depuis fin 1990 . Le chapitre ci-dessous ne teint pas encore compte de ces travaux .

L'administration de réseaux OSI repose sur trois modèles :

- un modèle fonctionnel
- un modèle organisationnel
- un modèle opérationnel

Le modèle fonctionnel reprend les grandes fonctionnalités déjà décrites :

- gestion des anomalies
- gestion de la comptabilité
- gestion de la sécurité
- gestion des performances
- gestion de la configuration et des noms

Comme nous le verrons, l'administration de réseaux OSI repose sur la notion d'objets gérés. Des standards décrivent aussi les fonctionnalités applicables à ces objets (voir ci-dessous).

Le modèle opérationnel ou d'information porte sur la description du réseau informatique en terme d'objets avec leurs caractéristiques, les opérations possibles sur ces objets et la manipulation des informations administratives à travers cette structuration en objets (attributs, méthodes (opérations, notifications, comportements), héritage, etc.)

Le modèle organisationnel reprend le découpage en domaines d'administration (par autorité, normes, géographique ou par organismes,...) et la hiérarchisation (partielle) de ces domaines.

Il décrit aussi la mise en oeuvre des opérations sur les objets grâce à

- une architecture du logiciel d'administration
- des protocoles d'administration

2.1 Modèle d'information

Il est articulé autour de la notion d'objet géré

2.1.1 Définition

Un objet est l'abstraction d'un composant physique ou logique . Il est caractérisé par
- un nom
- des attributs
- des opérations
- des notifications sur ces attributs

Appliqué à l'administration de réseaux on parlera d'objet géré, sous-ensemble des objets du système.

Le modèle d'information OSI est décrit dans le projet de standard DP 10165-1

On doit distinguer objet géré et ressource. L'objet géré est visible de l'administration de réseaux. Les ressources ne sont visibles qu'à la surface (boundary) de l'objet géré et traitées de manière interne. Seule cette partie visible est accessible.

Les objets sont regroupés en classes d'objets gérés. Seuls les aspects suivant sont visibles

-attributs visibles à la surface
- opérations applicables
- comportement en réponse à une opération
- notification émise par l'objet géré
- "packages" conditionnels encapsulés dans l'objet
- position de la classe d'objets dans la hiérarchie d'héritage
- spécification des objets allomorphes avec la classe (voir ci-dessous).

2.1.2 Principes

* Encapsulation

L'encapsulation assure l'intégrité d'un objet . Toutes les opérations passent par l'envoi d'un message à un objet . L'opération est interne et non visible sauf par la vue à la surface des attributs, notifications et opérations.

*** Héritage et classes d'objets**

Les instances d'un objet géré qui partagent les mêmes attributs, opérations, notifications, comportement et package font partie de la même classe.

Une classe peut être une extension d'une autre classe par ajout d'attributs, opérations, notifications etc.

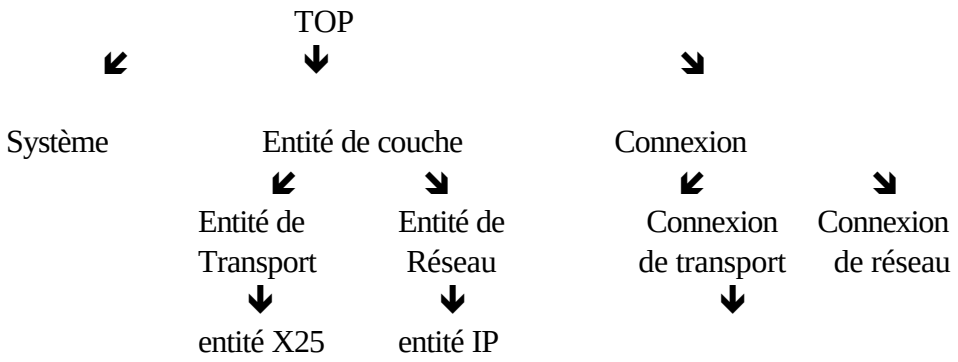
On définit ainsi des sous-classes de plus en plus spécialisées.

Une sous-classe hérite de tous les attributs, etc. de sa superclasse. La superclasse ultime est TOP; elle ne comporte aucun attribut ni autre propriété.

Un objet peut appartenir à plusieurs classes.

Une même sous-classe peut être spécialisée à partie de plusieurs superclasses (optionnel).

Exemple :



*** Allomorphisme**

L'allomorphisme représente la capacité pour une instance d'une sous-classe (sous-classe allomorphe) d'avoir un comportement comparable à celui de sa superclasse tel qu'observé par le protocole d'administration du système.

L'allomorphisme permet d'étendre la définition d'une classe d'objets pour permettre l'interopérabilité avec des administrations ou des objets gérés qui ne supportent pas l'extension de cette sous-classe. Ceci permet la migration des versions.

Cette extension peut se faire par

- addition d'attributs
- extension de portée des attributs
- restriction de portée des attributs
- ajout d'actions ou de notifications
- ajout d'arguments aux actions et notifications
- extension ou restriction sur la portée des arguments

Nous reviendrons plus loin sur les propriétés des objets et de leurs attributs.

2.2 Contenance et nommage

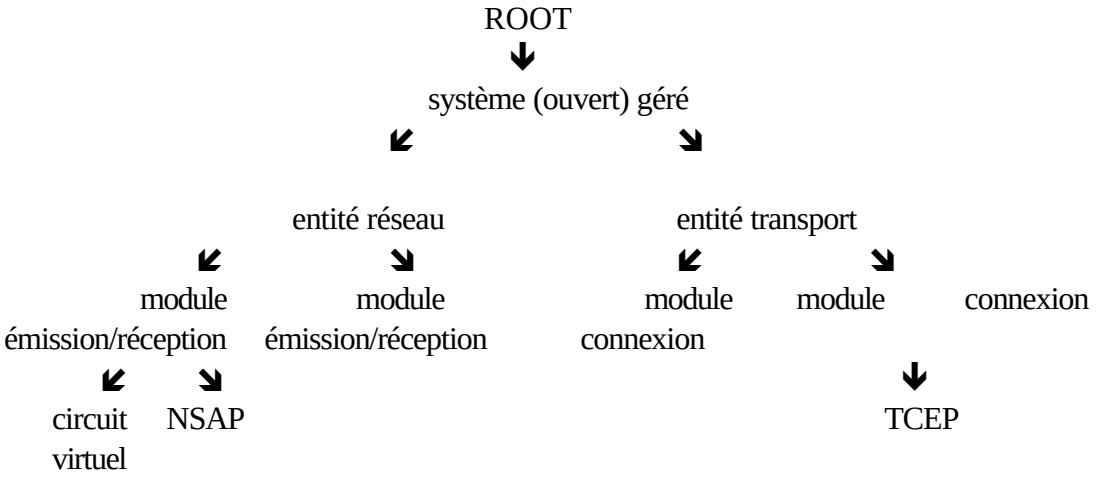
Un objet géré d'une classe peut contenir des objets gérés de la même ou d'autres classes . Cette relation est appelée contenance. Les objets contenus sont désignés comme étant des objets gérés subordonnés. On définit ainsi une seconde arborescence.

Un objet est subordonné à un objet supérieur. Le supérieur ultime est ROOT.

Cette relation de contenance permet de modéliser la hiérarchie du monde réel (assemblage de composants) ou une hiérarchie organisationnelle (répertoires, fichiers, enregistrements par exemple).

La relation de contenance peut définir un comportement statique ou dynamique de l'objet supérieur et de ses subordonnés.

Exemple :



Cet arbre de contenance est "orthogonal" à l'arbre hiérarchique. Il permet le nommage des objets.

Le nommage est hiérarchique. Un objet subordonné est nommé par :
- le nom de son supérieur

- un identificateur unique de subordonné dans la portée de contenance (du supérieur).

Ce nom peut être non ambigu dans un contexte de nommage local mais il lui peut être difficile de le rester dans des contextes très vastes. Il faut nommer de manière non ambiguë ces contextes (par exemple Domaine) et associer un nom de contexte.

Un objet géré ne peut exister que si son supérieur existe (créé et non détruit). L'objet ultime ROOT est un objet nul qui existe toujours.

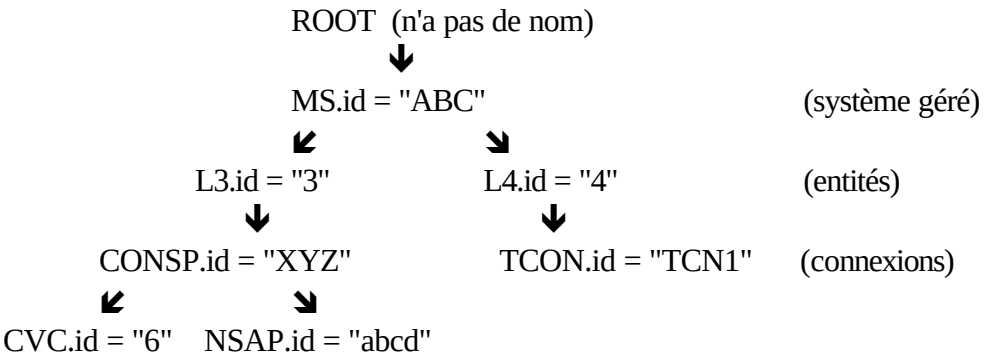
Les instances de classes d'objets gérés supérieurs qui peuvent être utilisées pour le nommage d'instances de classes d'objets sont identifiées en utilisant une "agrégation de noms" (names binding).

Ces régles de nommage constituent le schéma de nommage. On utilise deux types de noms :

- le nom relatif (relative distingueshed name)
- le nom absolu (global " ")

Une instance d'une classe d'objets gérés doit posséder au moins un attribut utilisable pour le nom relatif.

Exemple :



Noms relatifs	Noms absolus
MS.id = "ABC" L3.id = "3" CONSP.id = "XYZ" CVC.id = "6"	MS.id = "ABC" [MS.id="ABC",L3.id="3"] [MS.id="ABC",L3.id="3",CONSP.id="XYZ"] [MS.id="ABC",L3.id="3",CONSP.id="XYZ",CVC,id="6"]

Ainsi le nom global est bien un "agrégat" de noms relatifs.

Chaque attribut d'un objet géré est identifié par un identificateur, qui le distingue des autres attributs de l'objet. Cet identificateur est associé à l'attribut lui-même et non à son type.

2.3 Caractéristiques des objets

2.3.1 Attributs

Les attributs expriment les propriétés des objets gérés. Un attribut a une structure (ensemble ou séquence d'éléments) et prend une valeur particulière par une assertion de valeur d'attribut (AVA).

En général la valeur d'un attribut est observable (à la surface de l'objet); elle peut déterminer, à la suite d'une opération, ou refléter, par une notification, le comportement de l'objet.

Un attribut ne peut contenir ni un objet ni un autre attribut
Il est nommé de manière non ambiguë

Il est lu ou modifié de façon atomique par les opérations GET et SET (ensemble minimal) ou des opérations supplémentaires.

Toutes les opérations qui l'affectent sont faites de manière indirecte via l'objet contenant.
Si une opération porte sur plusieurs attributs, un système de synchronisation, géré par l'objet contenant, doit être mis en place.

L'attribut participe à l'héritage et à la contenance.

Il peut être obligatoire ou contenu dans un "package" conditionnel. Les attributs obligatoires sont toujours présents dans les instances des objets gérés d'une classe donnée.

Si un attribut est hérité d'une superclasse, il doit avoir un type abstrait qui possède le même ensemble d'opérations que celui de sa superclasse.

Il est possible de définir des groupes d'attributs.

*** Attributs de base**

Cinq attributs sont définis pour tous les objets gérés :

- Nom

Cet attribut permet au système gestionnaire de déterminer l'identificateur et la valeur du nom relatif de l'objet géré.

- Classe d'objet

identifie la classe actuelle de l'objet géré

- Superclasses allomorphes

identifie l'ensemble des superclasses allomorphes à la classe de l'objet.

- Chaînage de nom

identifie les classes d'objets gérés qui peuvent être nommées à partir de cet objet.

- package

identifie les packages qui ont été instanciés.

On trouvera ci-dessous une définition de cette caractéristique des objets gérés.

2.3.2 Opérations

Il existe 2 types d'opérations

- celles qui portent sur les attributs
- celles qui portent sur l'objet dans son ensemble.

Opérations sur les attributs :

GET lire la valeur; toujours applicable à un objet lisible.
REPLACE remplacer une valeur d'attribut (si il est inscriptible); ne s'applique pas aux groupes d'attributs.
SET to default remettre la valeur par défaut; s'applique à tous les attributs inscriptibles.
ADD member ajoute des membres fournis par l'opération à un attribut inscriptible et déjà positionné.
REMOVE member retire des membres à l'ensemble des membres d'un attribut positionné.

* Opérations globales :

CREATE crée et initialise un objet d'une classe spécifiée en utilisant la hiérarchie de nommage.
DELETE permet de supprimé un objet (s'il n'a plus d'objets subordonnés)
ACTION demande à un objet d'exécuter une action complexe donnée et, éventuellement, de lui en indiquer le résultat.

2.3.3 Notifications

Les objets gérés émettent "spontanément" des notifications lorsque des événements internes ou externes surviennent.

Ces notifications sont spécifiques de l'objet. Elles portent des informations définies dans l'objet.

2.3.4 Filtres

Les filtres permettent de spécifier des critères auxquels l'objet géré doit satisfaire pour qu'un opération soit exécutée.

Ils permettent la sélection d'objets multiples pour l'exécution d'opérations identiques sur ces objets.

Un filtre s'exprime en termes d'assertions et est satisfait seulement si la réponse à cette assertion est "TRUE". (Il a un comportement similaire à un prédicat)

Les opérateurs suivants sont utilisés par les filtres:
- and, or, not
- tests =, _, _, présent, sous chaîne de, sous-ensemble de, surensemble de, intersection d'ensembles non nulle

2.3.5 Comportement

Il définit :

- la sémantique des attributs, notifications, opérations.
- la réponse aux opérations invoquées
- les circonstances dans lesquelles les notifications sont émises
- les dépendances entre valeurs d'attributs particuliers
- les effets des relations sur les autres objets gérés.

2.3.6 Packages conditionnels

Un package (conditionnel) est une collection d'attributs optionnels, de notifications, d'opérations et de comportements qui sont tous présents ou absents simultanément. Cette présence (ou absence) est conditionnée par les capacités des ressources de niveau inférieur (par exemple : options dans un protocole de communication).

Pour un tel package une seule instance peut exister. Il n'est dons pas besoin d'un chaînage de nom.

Il ne peut être instancié que si il est encapsulé; cette instanciation est réalisée avec l'objet (les opérations passent toujours par l'objet et ne s'appliquent pas directement au package .

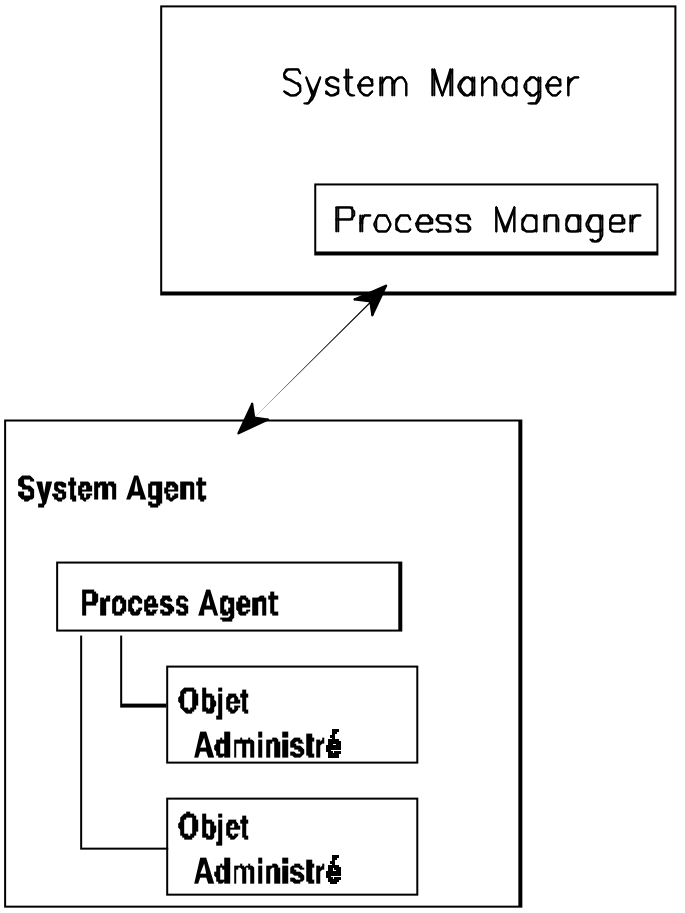
2.4 ARCHITECTURE DE L'ADMINISTRATION OSI

L'administration de réseaux OSI supporte la partition en domaine. Elle est donc répartie sur des systèmes gestionnaires et des systèmes gérés.

Dans les systèmes gérés un processus agent traite les objets gérés.

Dans les systèmes gestionnaires on trouve aussi un processus agent pour administrer les objets locaux. Un processus gestionnaire administre l'ensemble du domaine.

Les processus agents réalisent les opérations sur les objets et à travers eux sur leurs attributs. Ils transmettent les notifications.



2.4.1 Structure de la gestion de réseau OSI

La gestion de réseau OSI est réalisée par

- la gestion-système
- la gestion de couche (N)
- les opérations de couche (N)

Les opérations de couche (N) sont intégrées aux entités de communication de niveau (N).

La gestion de couche (N) est réalisée par une entité d'administration spécifique placée au niveau (N) à coté des entités de communication de ces niveaux.

La gestion-système est réalisée par des entités de la couches Application (niveau 7 OSI).

Les protocoles de gestion de couche ne devraient être utilisés que si des besoins spéciaux rendent inappropriés les protocoles de gestion-système ou si ils ne sont pas disponibles. C'est en particulier le cas pour les commutateurs de paquets ou les ponts de niveau 3 OSI; sur ces systèmes on ne peut installer de gestion-système au niveau 7. Les problèmes d'acheminement qui relèvent de l'administration de réseaux et qui sont mis en oeuvre dans ces équipements relèvent dont de la gestion de couche ou d'opérations de couche.

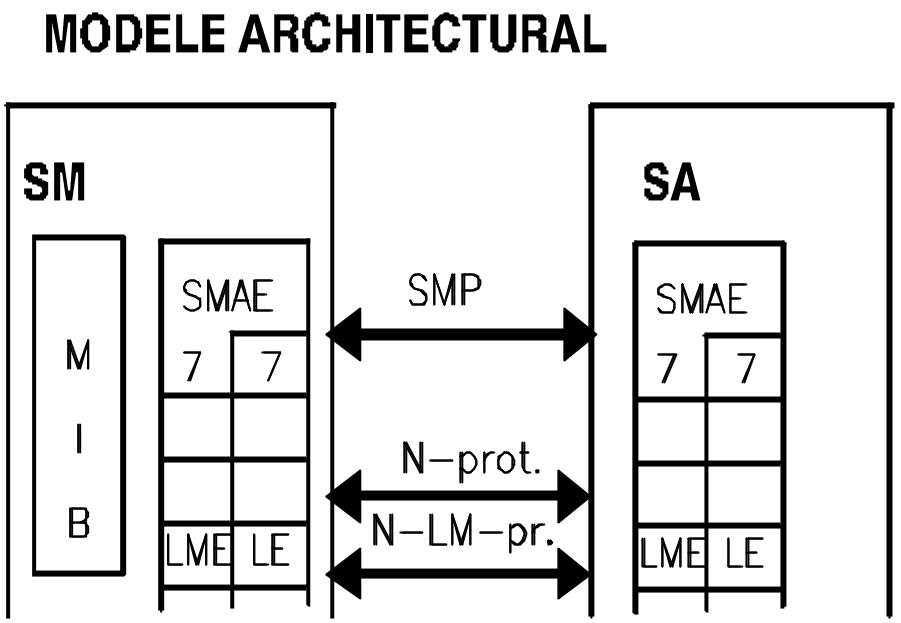
Les opérations de couche peuvent exister dans les 7 couches du modèle de Référence. Les informations transportées doivent être distinguables des données utilisateur. Cette distinction incombe au protocole de niveau N. Par exemple des paramètres de taxation sont transportés au niveau 3 OSI dans des champs de facilités.

Ces informations ont pour but de commander er de surveiller une instance de communication unique . Elles comportent :

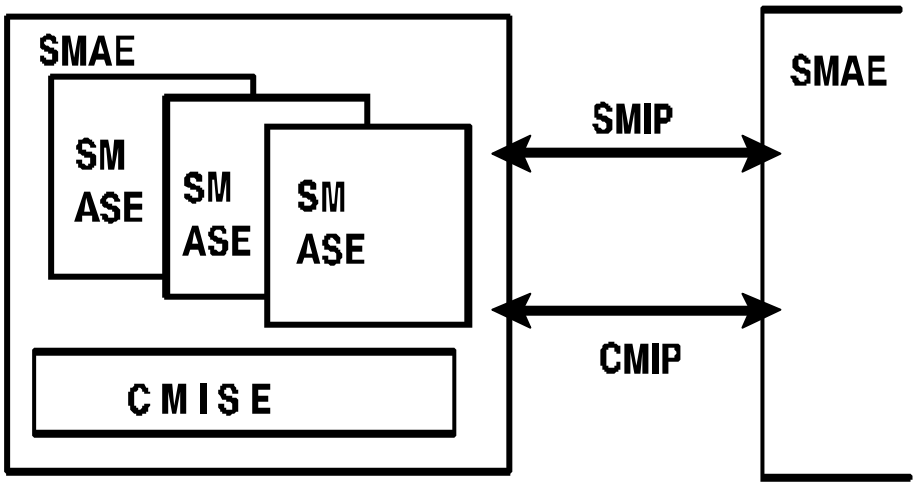
- des paramètres dans les PDU de connexion ou d'association
- des paramètres de PDU particulières qui peuvent modifier le comportement de cette instance de communication
- des informations d'anomalies
- des paramètres des PDU de terminaison ou de rupture d'association relatifs à l'instance qui se termine.

L'essentiel de l'administration de réseaux est traité par la gestion système.

Les schémas ci-dessous illustrent l'architecture de l'administration de réseaux PSI. L'architecture de la gestion-système suit les normes de l'ALS (architecture de la couche Application). Elle comporte un service commun : CMIS et des entités spécifiques d'administration (SMAE).



SMAE : System management Application Entity



SMIP : System management Information Protocol
CMIP : Common Management Information Protocol
SM ASE : System Management Service Element

2.4.2 Base de données de gestion : MIB

Dans chaque système, une MIB représente les informations qui peuvent être transférées par les protocoles de gestion OSI (gestion système, gestion de couches) ou qui sont concernées par l'utilisation de ces protocoles.

La MIB est constituée par l'ensemble des objets gérés dans un système ouvert. Seuls les objets de l'environnement OSI entrent dans le cadre de la normalisation,

qui ne s'applique qu'à leur structure logique. Cependant la MIB peut contenir d'autres objets.

Ceci ne suppose rien non plus quant à la forme de stockage physique ou logique dont la réalisation est locale et propre à chaque système.

Les informations de gestion peuvent être partagées ou structurées suivant les besoins des processus de gestion. Elles peuvent être stockées à l'état brut ou après traitement.

2.4.3 Gestion système

Elle fournit les mécanismes nécessaires à la surveillance, au contrôle et à la coordination des objets de gestion par l'utilisation de protocoles dans la couche Application :

- par des entités d'application de gestion-système :SMAE
- par une entité commune d'application : CMISE

Le logiciel de communication doit offrir les fonctionnalités suffisantes pour supporter CMISE et les SMAE (sinon on ne peut utiliser que les systèmes de gestion de couche pour les couches supportées).

La majorité des échanges d'information de gestion entre systèmes ouverts nécessite la négociation d'un contexte de présentation particulier et l'établissement d'une session supportée par un transport fiable. Elle est donc supportée par la couche Application. ceci n'exclut pas un service en mode sans connexion.

2.4.4 Gestion de couche (N)

Elle assure, si nécessaire, la surveillance, le contrôle et la coordination des objets de la couche (N).

Ses protocoles sont pris en charge par le service (N-1). Elle ne fournit aucun service au niveau (N+1).

Ces protocoles assurent :

- la communication de valeurs de paramètres liés aux objets de la couche (N)
- le test des fonctions fournies par le service (N-1)
- le transport d'informations d'erreur pour gérer les anomalies et les diagnostics.

2.5 Le service commun d'administration OSI: CMISE

Ce service commun, inclus dans la couche Application fournit aux SMAE un certains nombre de fonctions; pour les fonctions d'association et de rupture, ces entités doivent faire appel au service ACSE, par les primitives AASSOCIATE, A-RELEASE, A-ABORT.

CMISE fournit 2 types principaux de transfert d'information :

- un service de notifications de gestion
- un service d'opérations de gestion

plus 2 services additionnels

- un service de réponses multiples pour confirmer les réponses qui lui seront liées

- un service d'opérations multiples sur des objets gérés multiples qui doivent satisfaire à des critères communs et sont sujet à des conditions de synchronisation.

Ces services sont :

notifications
M-EVENT-REPORT confirmé ou non

opérations
M-GET confirmé
M-SET confirmé ou non
M-ACTION confirmé ou non
M-CREATE confirmé
M-DELETE confirmé

Les services M-CREATE et M-DELETE agissent sur les objets globaux; elles servent à les créer et les supprimer.

Les autres services affectent les attributs des objets via ceux-ci.

M-EVENT-REPORT rapporte un événement affectant un objet
M-GET demande des informations de gestion à une entité paire
M-SET modifie des informations de gestion sur une entité paire
M-ACTION demande l'exécution d'une action complexe à une entité paire.

Les noms des instances des objets gérés sont organisés hiérarchiquement selon un arbre d'informations de gestion.

Des opérateurs de filtrage permettent de tester la présence ou la valeur d'attributs.

Un paramètre de synchronisation est fourni pour permettre d'indiquer la manière de synchroniser les opérations dans une instance d'un objet géré, lorsque des opérations multiples sont sélectionnées par filtre.
Deux types de synchronisation sont supportées :

- atomique
- effort maximal (best effort)

CMISE est organisé en unités fonctionnelles :

- une unité fonctionnelle noyau traite tous les services de base listés ci-dessus.

- des unités fonctionnelles additionnelle permettent d'étendre le service :

* sélection d'objets multiples
* filtre
* réponses multiples
* service étendu (qui permet d'accéder au service P_DATA de la couche Présentation)

Les deux exemples ci-dessous, donnés à titre indicatif, montrent des primitives associées aux services M-EVENT-REPORT et M-CREATE et leurs paramètres.

notations : Rq/Ind requête ou indication
Resp/Conf réponse ou confirmation

M obligatoire
U utilisateur
C conditionnel
= identique à la requête

M-EVENT- REPORT		Req/Ind	Resp/Conf
invoke identifier		M	M=
mode			M
classe d'objet géré		M	U
instance d'objet géré		M	U
type	d'événement	M	C=
instant de l'événement		U	
info. sur l'événement		U	
temps courant			U
réponse à l'événement			C
erreurs			C

M-CREATE	Req/Ind	Resp/Conf
invoke identifier	M	M=
classe d'objet géré	M	C
instance d'objet géré	U	C
instance d'objet supérieur	U	
contrôle d'accès	U	
instances d'objets inférieurs	U	
liste d'attributs	U	C
temps courant		U
erreurs		C

2.6 Protocole CMIP

Le service CMISE est rendu par des entités d'Application qui mettent en oeuvre le protocole CMIP : Common Management Information Protocol.

Ce protocole utilisent les services Application
- ACSE
- ROSE

et le service Présentation P-DATA

Le service ACSE est mis en oeuvre pour établir et rompre les associations. L'association utilisée est de classe 3.

Le service ROSE est utilisé pour les opérations et les notifications. Les services
- RO-Invoke
- RO-Result
- RO-error
- RO-Reject

sont mis en oeuvre. Les opérations confirmées sont de classe 2 : asynchrone (retour immédiat) ou 1 : synchrone (retour en fin d'échange). Les opérations non confirmées sont de classe 5 : synchrone avec sortie non rapportée.

CMIP définit donc un ensemble d'opérations qui seront transmises pour exécution par ROSE.

Chaque opération est décrite de ASN1.

Elle est reliée à une primitive CMISE et soumise à ROSE par RO-Invoke (opération).

Exemple : Description du service M-EVENT-REPORT

Procédure.

Sur réception de la primitive M-EVENT-REPORT, la machine CMIP doit :

* en mode confirmé, émettre une APDU demandant l'opération m-EventReport-Confirmed ou

* en mode non confirmé, émettre une APDU demandant l'opération m-EventReport

envoyer cette opération en utilisant la procédure RO-INVOKE

A la réception de cette APDU, la machine CMIP émet une indication M-EVENT-REPORT à l'utilisateur de CMISE.

En mode confirmé, cet utilisateur fournit une réponse à partie de laquelle la machine CMIP

construit une APDU confirmant cette notification

si les paramètre de la réponse montrent que la notification a été acceptée, émet cette APDU en utilisant le service RO-RESULT (de ROSE); sinon elle émet cette APDU par le service RO-ERROR (de ROSE).

Quand la machine CMIP initiatrice reçoit cette APDU elle transmet, si elle est bien formée, la confirmation correspondante à l'utilisateur de CMISE. Sinon elle construit une APDU contenant notification de l'erreur et l'envoie en utilisant RO-REJECT-U.

Structure des APDU m-EventReport
et m-EvenrReport-Confirmed

```
IMPORTS
  OPERATION, ERROR FROM Remote-Operation-Notation

  DistinguishedName, RDNSequence FROM
                                        InformationFramework


m-EventReport OPERATION
 ARGUMENT        EventReportArgument
 ::= localValue 0

m-EventReport-Confirmed     OPERATION
 ARGUMENT        EventReportArgument
 RESULT          EventReportResult      -- optionnel
 ERRORS {
      invalidArgumentValue,noSuchArgument,noSuchEventType,noSuchObjectClass,no
SuchObjectInstance,processingFailure }
 ::= localValue 1


invalidArgumentValue  ERROR
 PARAMETER       InvalidArgumentValue
 ::= localValue 15

noSuchArgument                  ERROR
```

```
 PARAMETER        NoSuchArgument
 ::= localValue 14

noSuchEventType              ERROR
 PARAMETER        NoSuchEventType
 ::= localValue 13

noSuchObjectClass            ERROR
 PARAMETER        ObjectClass
 ::= localValue 0

noSuchObjectInstance  ERROR
 PARAMETER        ObjectInstance
 ::= localValue 1

processinfFailure            ERROR
 PARAMETER        ProcessingFailue  -- optionnel
 ::= localValue 10


EventReply      ::= SEQUENCE {
        eventType               EventTypeId,
eventReplyInfo [8]          ANY          DEFINED          BY          eventType
                                                        OPTIONAL }

EventReportArgument  ::= SEQUENCE {
        managedObjectClass          ObjectClass,
managedObjectInstance       ObjectInstance,
eventTime       [5] IMPLICIT GeneralizedTime
                                                              OPTIONAL,
        eventType       EventTypeId,
eventInfo       [8] ANY DEFINED BY eventType
                        OPTIONAL }

EventReportResult       ::= SEQUENCE {
        managedObjectClass                      ObjectClass       OPTIONAL
managedObjectInstance                 ObjectInstance                OPTIONAL
currentTime             [5]IMPLICIT          GeneralizedTime          OPTIONAL
eventReply      EventReply OPTIONAL}


EventTypeID   ::= CHOICE {
        globalForm              [6]       IMPLICIT       OBJECT       IDENTIFIER
localForm       [7] IMPLICIT INTEGER }

InvalidArgumentValue  ::= CHOICE {
```

```
        actionValue             [0]     IMPLICIT        ActionInfo
eventValue  [1] IMPLICIT SEQUENCE {
        eventType  EventTypeId
        eventInfo  [8] ANY DEFINED BY eventType OPTIONAL }}

NoSuchArgument          ::= CHOICE {
        actionId  [0] IMPLICIT SEQUENCE {
        managedObjectClass    ObjectClass OPTIONAL,
        actionType            ActionTypeId }
        eventId [1] IMPLICIT SEQUENCE {
        managedObjectClass    ObjectClass OPTIONAL,
                eventType                   EventTypeId }}

NoSuchEventType ::= SEQUENCE {
        managedObjectClass    ObjectClass                   ,
        eventType                   EventTypeId }

ObjectClass     ::= CHOICE {
        globalForm              [0]     IMPLICIT        OBJECT      IDENTIFIER
localForm       [1] IMPLICIT INTEGER }

ObjectInstance ::= CHOICE {
        distinguishedName       [2] IMPLICIT DistinguishedName,
        nonSpecificForm         [3] IMPLICIT OCTET STRING,
        localDistinguishedName [4] IMPLICIT RDN Sequence}

ProcessingFailure       ::= SEQUENCE {
        managedObjectClass    ObjectClass,
managedObjectInstance         ObjectInstance OPTIONAL
        specificErrorInfo       [5] ANY DEFINED BY
                                        managedObjectClass }
```

2.7 Gestion-Systeme

Nous n'étudierons pas en détail tous les aspects de la gestion-système.

La description des services correspondants est en cours de normalisation et réalisée dans les divers documents du projet de standard 10164-x.

Les structures de données correspondant aux protocoles sont fournies dans les projets de standard 10165-x.

2.7.1 Fonctions (actuellement) supportées

Neuf fonctionnalités sont en cours de spécification. Cinq d'entre elles sont déjà publiées :

- Gestion d'objets
- Gestion d'état
- Gestion des relations
- Compte-rendu d'alarmes
- Compte-rendu d'événements

- Contrôle d'archivage (log)
- Compte-rendu d'alarmes de sécurité
- Synthèse de mesures
- Synthèse de diagnostics

Nous étudierons plus particulièrement les trois premières.

2.7.2 Gestion d'objets

Il s'agit d'une fonctionnalité "passe tout droit" (pass-through).

Elle comporte 6 primitives :

- P-create
- P-delete
- P-action
- P-set
  replace
add
remove
set-to-default
- P-get

- P-event (notifications)

Les objets gérés suivent le modèle d'information décrits plus haut avec leurs caractéristiques

- attributs
- opérations
- comportements
- notifications
- packages conditionnels
- position dans la hiérarchie d'héritage
- allomorphisme

2.7.3 Gestion d'états

La gestion d'états traite de la disponibilité des objets du point de vue de leur :
opérabilité
usage
administration

Ces états sont consignés dans deux attributs :
état opérationnel (opérabilité et usage)
état administratif

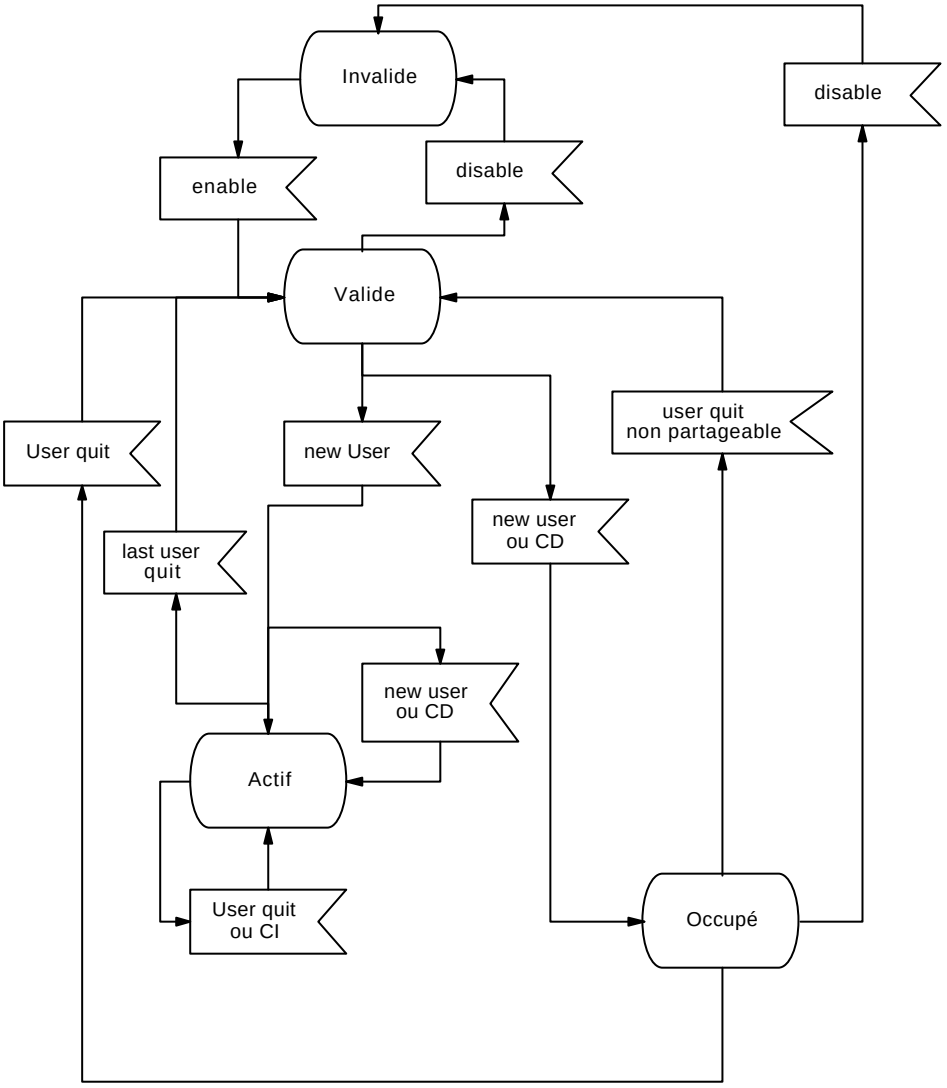
L'ensemble de ces deux attributs constitue l'état de gestion.

Etat opérationnel

Il peut prendre 4 valeurs :

- invalide (opérabilité)
- valide
- actif (usage)
- occupé

Ces états sont "read-only". Ils ne peuvent pas être modifiés par l'administration de réseaux.

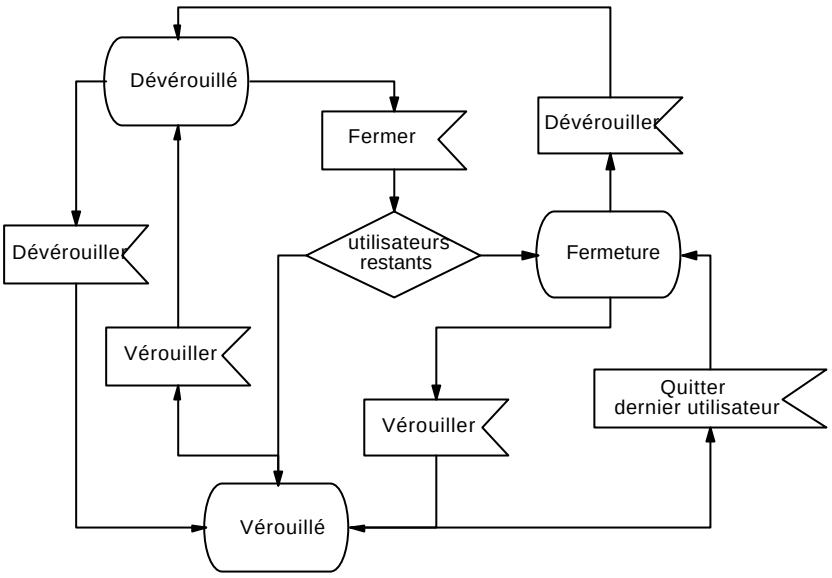


CD capacité décroît
CI capacité croît

Etat administratifs

Il peut prendre 3 valeurs :

- verrouillé
- déverrouillé
- fermeture (en cours)



Etat de gestion

Combinaison des deux états précédents, il peut prendre 10 valeurs. Les états actif-verrouillé et occupé-verrouillé sont interdits. Deux états invalide-fermeture et valide-fermeture sont temporaires et ont une transition spontanée vers les états respectifs invalide-verrouillé et valide-verrouillé.

opérationnel administratif	INVALIDE	VALIDE	ACTIF	OCCUPE
VEROUILLE	Invalide Vérouillé	Valide Vérouillé	impossible	impossible
FERMETURE	basculement automatique vers invalide occupé	basculement automatique vers valide occupé	Actif Fermeture	Occupé Fermeture
DEVEROUILLE	Invalide Dévérouillé	Valide Dévérouillé	Actif Dévérouillé	Occupé Dévérouillé

"Santé" (Health)

Un attribut "santé" contient des informations plus détaillées sur cet état de gestion; il peut prendre les valeurs :

- anomalie rapportée
- en faute
- en réparation
- réservé pour test
- en tests
- non installé
- jamais installé
- jamais utilisé
- hors tension
- hors ligne
- hors usage (après un temps limite)
- initialisation incomplète
- initialisation requise

Un attribut groupe d'états contient l'état de gestion et la santé de l'objet.

2.7.4 Gestion des relations

Une relation (relationship) décrit comment une opération portant sur une partie d'un système affecte les opérations d'un autre système.

Cette relation peut être
directe
ou indirecte (via un objet intermédiaire)

Elle peut être aussi
symétrique (influence identique dans les 2 sens)
asymétrique (rôle des objets différents)

Catégories de relations

- Contenance

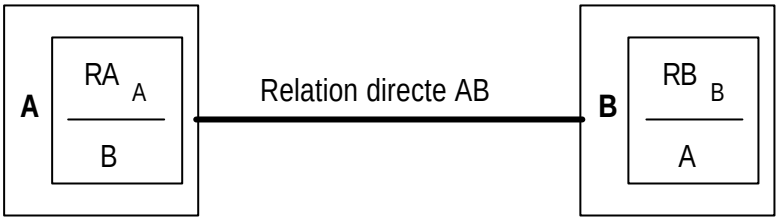
Ce type de relation est créé automatiquement à la création d'un objet.

Elle est utilisée notamment pour la suppression des objets subordonnés : Il n'est possible de supprimer un objet supérieur que si tous les objets subordonnés ont été détruits.

exemple : répertoire / fichiers)

- Relation explicite

Une telle relation crée un lien entre 2 objets. Un attribut contient le nom de l'autre objet (attribut relation RA , ici RA_A contient B)



Les relations explicites sont

crées	par Add
supprimées	par Remove
mofifiées	par Replace

La création d'un objet implique la création de ses relations explicites par des attributs que l'on trouve dans create.

Les informations sur les relations explicites peuvent être lues par une opération read.

Une relation explicite peut être unidirectionnelle . Dans ce cas l'attribut relation contenant le nom est dans un seul objet.

- Objets gérés qui représentent des relations

Dans une relation indirecte, l'objet intermédiaire représente la relation indirecte. Cet objet intermédiaire est suffisant pour désigner la relation indirecte (avec ses attributs de relation directe).

Types de relations explicites

- relation de service
  fournisseur / client d'un service

- relation paire (peer)
  entre 2 entités de communication distantes

- relation de repli (fallback)
  primaire / secondaire avec second choix préféré

- relation de remplacement (backup)
  remplaçant / remplacé, si un objet est à l'état invalide

- relation de groupe
  propriétaire / membre

Rôles des relations

Les relations peuvent avoir des rôles identiques ou complémentaires. Dans le premier cas, la relation est symétrique, dans le second elle est asymétrique. Les exemples ci-dessus donnent des exemples de rôles connus.

2.7.5 Compte-rendu d'alarme

Ils portent sur différents types d'alarmes
- communications
- qualité de service
- traitements
- équipements

Il est possible de signaler une cause probable pour chaque type, la sévérité perçue, la tendance (alarme plus sévère, moins sévère ou sans changement) et le remplacement d'un objet en alarme.

Causes probables (exemples)

Communications :
perte du signal
erreur trame
erreur de transmission locale
erreur de transmission distante
erreur d'établissement de connexion

Qualité de service
  temps de réponse excessif
  débordement de queue

Traitements
  capacité de stockage dépassée
  erreur de version etc.

environnement
incendie - fumées
humidité excessive
température excessive etc.

Sévérité perçue

- clair (cleared) après prise en compte
- indéterminée
- critique
- majeure
- mineure
- attention (warning)

Informations de seuil
- valeur observée
- seuil de déclenchement de l'alarme
- niveaux de seuil en cas d'hystérésis
- temps d'armement (depuis le dernier réarmement)

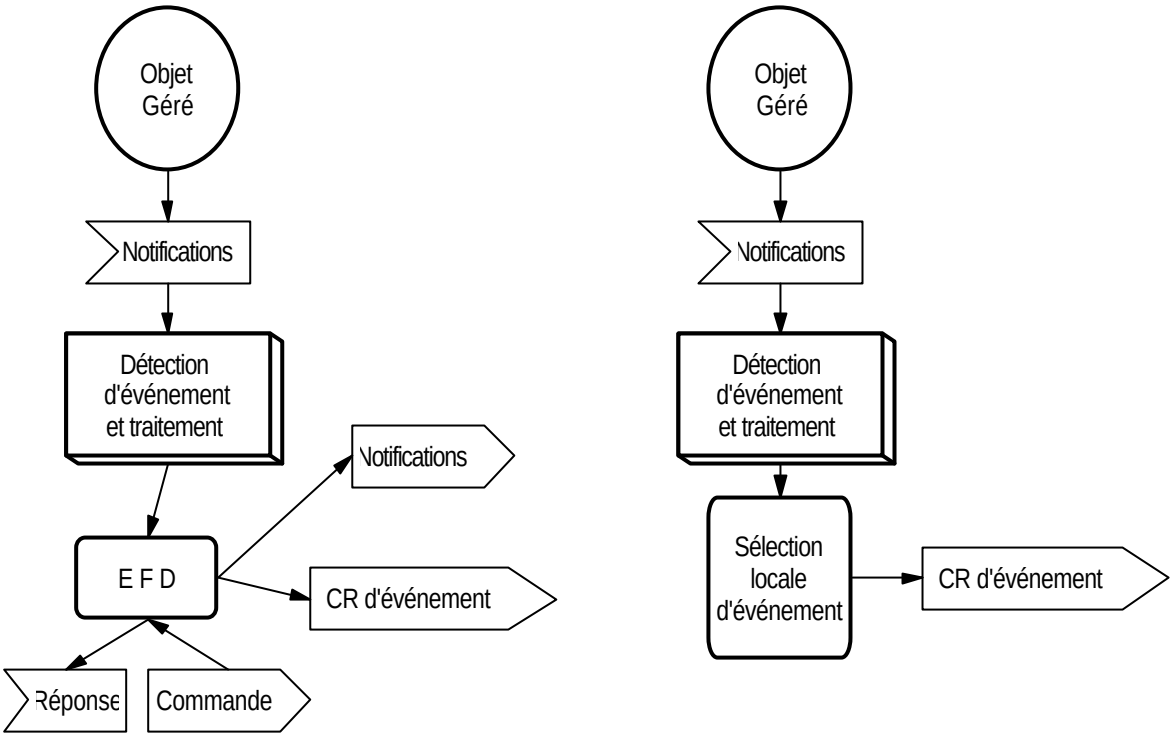
2.7.6 Compte-rendu d'événements

Cette fonctionnalité permet de traiter et de rapporter à d'autres systèmes les notifications d'événements.

L'événement notifié est traité localement et/ou rapporté à un autre système (pour traitement local ou rapport à distance) . Ceci est réalisé soit par notification (autre événement) soit par un compte-rendu d'événement.

Modèle

Ce modèle décrit les composants (conceptuels) qui fournissent le compte-rendu d'événements distants et leur traitement local.



CR : Compte rendu
EFD : Event Forwarding Discriminator

Le "filtre" (discriminator) d'événement est un support d'objets gérés qui permet à un "gestionnaire" de contrôler les opérations de gestion et les compte-rendu d'événement relatés par d'autres objets gérés.

3. Base de données administratives : MIB

Administration de réseaux SNMP

Actuellement l'Administration de Réseaux hétérogènes s'organise essentiellement autour du protocole SNMP (Simple Network Management Protocol) développé dans le cadre de l'architecture Inet (TCP/IP).

3.1. MIB pour INTERNET : Objets Gérés

Ils sont décrits par seulement 5 paramètres :
Descripteur de l'objet
Nom et identificateur dans l'arbre de nommage
Syntaxe
Type en ASN.1

Définition
Description en texte libre
Droits d' Accès
lecture seule
lecture - écriture
écriture seule
non accessible
Statut
obligatoire
optionnel
obsolète
dépréciée (deprecated)

Ainsi un objet est caractérisé par
un NOM
une SYNTAXE
un CODAGE

La syntaxe utilisée est un SOUS_ENSEMBLE de ASN.1.La structure est une version très simplifiée de la structure des objets OSI avec seulement 2 ATTRIBUTS en plus du nom.

3.2. Syntaxe ASN.1 pour MIB INTERNET

Le sous-ensemble utilisé comporte les tyes suivants :

INTEGER		SEQUENCE { }	
OCTET STRING		SEQUENCE OF	
OBJECT IDENTIFIER		NULL	

et les types Applications :

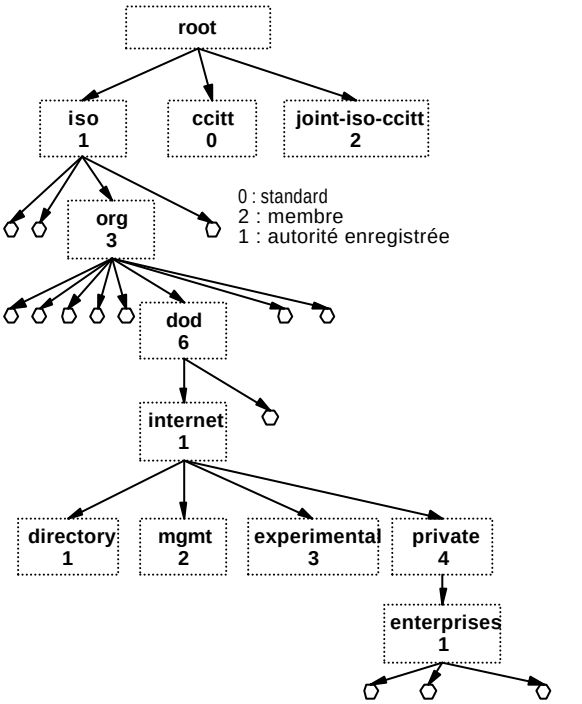
NetworkAdress			(choix { internet IpAdress})
IpAdress			(chaine de 4 octets)
Counter			(entier 0..4294967295)
Gauge	Seuils		(entier 0..4294967295)
Timeticks		Temporisation	(entier 0..4294967295)
Opaque		Masque tout type spécifique	(chaine d'octets)

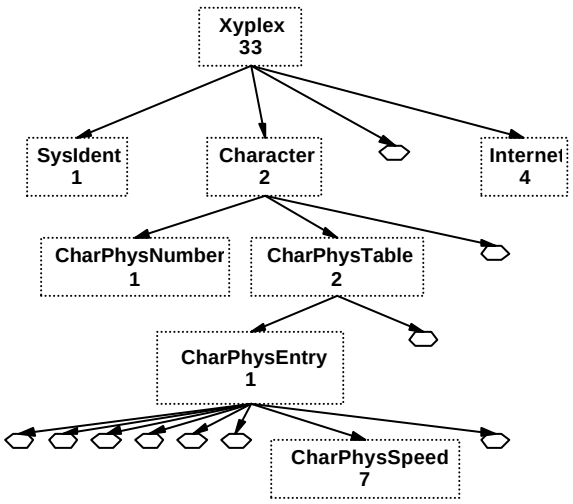
Exemple issu de la MIB II :

OBJECT :
IpAddrTable {ip 20}
Syntax :
SEQUENCE OF IpAddrEntry
Definition:
The table of adressing information relevant to this entity's IP adresses
Acces :
read-only
Status :
mandatory

3.3. Arbre de nommage

Il est défini par l'OSI et utilisé par toutes les autres organisations en particulier INTERNET pour sa MIB





3.4. Exemple de description d'un objet

Débit d'un port d'un concentrateur de terminaux Xyplex pour SNMP

entreprise XYPLEX

1.3.1.6.1.4.1.33

OBJECT

SysIdent {system 1}
Syntax : DisplayString (SIZE (0..40))
Definition : (chaîne d'identification .....)
Acces : read-only
Status : Mandatory

OBJECT

charPhysNumber {character 1}
Syntax : INTEGER
Definition : (Nombre de ports physiques .....)
Acces : read-only
Status : Mandatory

OBJECT

charPhysTable {character 2}
Syntax : INTEGER
Definition : (Liste des ports d'entrée arytmiques.....)
Acces : read-only
Status : Mandatory

OBJECT

charPhysEntry {charPhysTable 1}
Syntax : charPhysEntry ::= SEQUENCE {
charPhysIndex,
INTEGER,
charPhysSpeed,
INTEGER,
}
Definition : (Caractéristique d'un port d'entrée .....)
Acces : read-only
Status : Mandatory

OBJECT

charPhysSpeed {charPhysEntry 7}
Syntax : INTEGER
Definition : (vitesse du port.)
Acces : read-write
Status : Mandatory

Ainsi la vitesse d'un port de ce concentrateur est un objet administrable qui peut être lu ou positionné. Il est identifié de manière unique par

charPhysSpeed OBJECT IDENTIFIER ::=
{org dod internet private enterprises XYPLEX
character charPhysTable charPhysEntry 7}

soit 1.3.6.1.4.1.33.2.2.1.7

3.5. MIB-II

Elle suit la RFC 1158

3.5.1. Contenu

L'objet MIB-II correspond au noeud 1.3.6.1.2.1 dans l'arbre de nommage

Noeud	Type	Nombre	Commentaires
1	system	7	Noeud administré
2	interface	23	Attachement réseau (Ethernet, TokenRing, X25, etc)
3	at (ARP)	3	Translation de l'adresse IP en adresse niveau 2/OSI
4	IP	38	Protocole IP (niveau 3)
5	ICMP	26	Protocole ICMP (niveau 3 - supervision)
6	TCP	19	Protocole de Transport avec connexion TCP
7	UDP	7	Protocole de Transport sans connexion UDP
8	EGP	18	Routeurs (Gateway)
9	transmission	0	Objets réseau spécifique
10	SNMP	30	Niveau 7 - Administration de réseaux SNMP

Ainsi le groupe IP correspond au noeud 1.3.6.1.2.1.4

3.5.2. Exemple

systemGroup
system OBJECT IDENTIFIER :: { mib 1 }
sysDescript : description de l'appareil
sysObjectId : identificateur de l'appareil
(N° d'objet dans MIB privée 1.3.2.45....)
sysUpTime : durée (en seconde) depuis laquelle l'appareil est actif
sysContact : nom de la personne à contacter pour cet appareil
(ex: adm@ifhpserv.insa-lyon.fr)
sysName : nom de l'appareil (ifpc56)
sysLocation : localisation de l'appareil (batiment 501 T202)
sysServices : services offerts ; code les niveaux OSI par $\Sigma\ 2^{L-1}$
(ex : transport + administration = 8+64 = 48 hexa)

3.6. Opérations SNMP

3.6.1. Types

Chaque équipement géré est vu comme un ensemble de variables que l'on peut:

consulter Get-Request

Get-Next-Request
Table "Traversal" : Traversée pour parcourir des tables (en particulier les tables de routage)

modifier Set-Request

à distance. Cet ensemble correspond à un objet dans le modèle d'information OSI, les variables étant des attributs.

On peut aussi en recevoir spontanement des informations (notifications) :

trap

3.6.2. Le "puissant" GET-NEXT

La primitive Get-Next permet les opérations de type "Traversal" pour explorer les tables. Il est en particulier utilisé pour les tables d'adresses ou les tables de routage; il permet aussi de trouver des objets dans une liste.

Le problème à résoudre est le suivant: comment trouver une valeur dans une table de la MIB avec SNMP qui ne traite que des objets simples (contrairement à l'Administration OSI qui traite les listes d'attributs).

Si par exemple on présente la requête "get-next (sysDescript)
on obtient la valeur de l'objet suivant soit : sysObjectId

Cette primitive est donc utile pour traiter des tables dont on ne connait pas le contenu réel ou la taille.

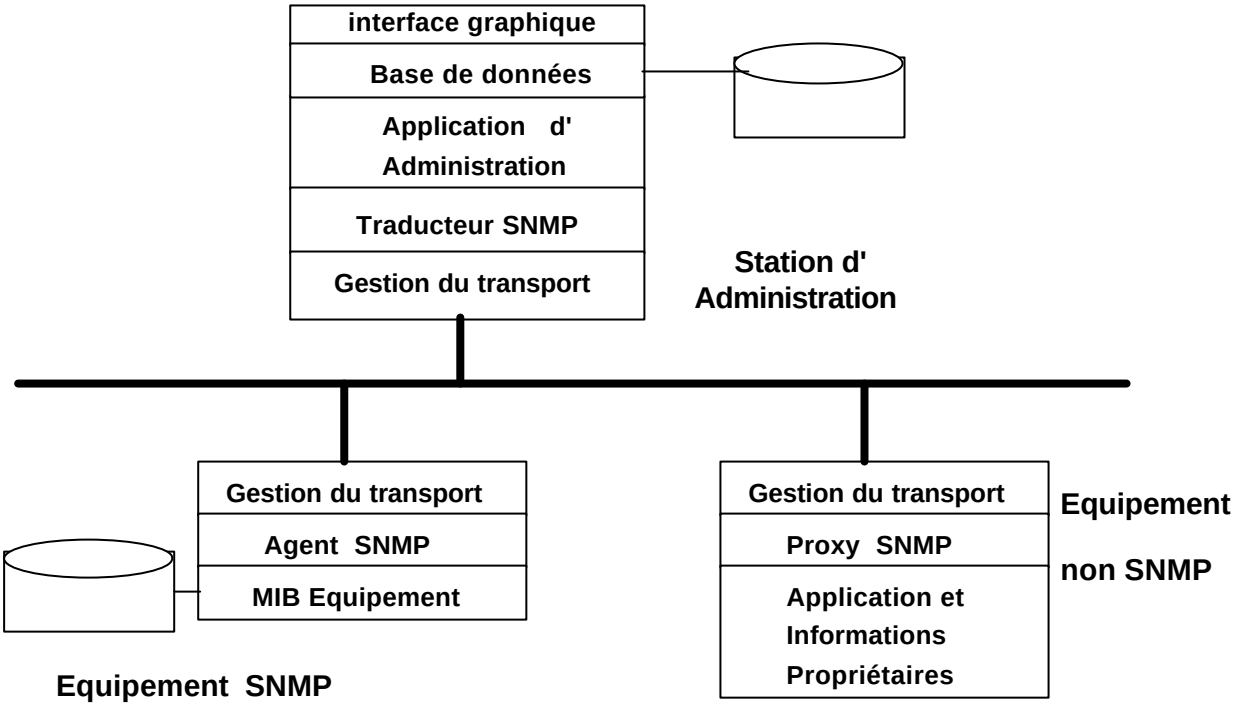
Get-next peut avoir plusieurs paramètres (comme set ou get) pour explorer les tables à plusieurs colonnes.

Exemple : get-next (ipRouteDest, ipRouteIfIndex, ipRouteNextHop)
donne la première ligne de la table de routage
si on rappelle get-next avec la valeur reçue on obtient la ligne suivante dans la table de routage.

Rq: ipNextHop identifie le routeur suivant dans le réseau; il peut ainsi être testé à son tour.

3.7. Architecture SNMP

3.7.1. Architecture : stations SNMP et Proxy



3.7.2.Equipements administrés

systèmes hôtes : stations de travail, serveurs
concentrateurs de terminaux
imprimantes
etc

routeurs, passerelles

Equipements de transmission :Ponts, multiplexeurs

Les messages sont transportés par des datagrammes UDP (port 161)

3.8. Messages SNMP

Ces messages sont codés dans un sous-ensemble de la syntaxe ASN.1.
Un message est codé par une séquence ASN.1 :

message ::SEQUENCE { version, community, data}

La version 1 correspond au standard RFC 1157
La communauté constitue le mécanisme d'authentification de SNMP; ce paramètre agit comme un mode de passe transmis du l'administrateur client à un agent administré.
Le champ data contient une PDU correspondant à une des requêtes ou trap.

Requètes :

GetRequest-PDU
GetResponse-PDU
GetNextRequest-PDU
SetRequest-PDU

Paramètres :
identificateur
type d'erreur
noError (0)
tooBig (1)
noSuchName (2)
BadValue (3)
readOnly (4)
genErr (5)
index d'erreur (numéro de la variable en erreur)
liste (agrégat) de variables (nom et valeur)

Trap :
Trap-PDU
Paramètres :
identification de l'agent source
adresse de l'agent
type de trap
coldStart (0)
warmStart (1)
linkDown (2)
linkUp (3)
authentificationFailure (4)
egpNeighborLoss (5)
enterpriseSpecific (6)
champ spécifique pour traps particuliers
date de l'événement

liste des variables associées à ce trap