

CMIS / CMIP

Architecture, Protocole et Services

EFORT

<http://www.efort.com>

L'ISO a proposé dans les années 80 la norme CMIS/CMIP (Common Management Information Service, Common Management Information Protocol) comme protocole d'administration de réseau utilisé entre un gestionnaire et un agent. Le but de ce tutoriel est de décrire l'architecture, le protocole et les services CMIS/CMIP.

1 Le modèle gestionnaire-agent

La gestion OSI (Open System Interconnection) fait reposer son modèle relationnel sur deux concepts importants qui sont le gestionnaire et l'agent, et précise aussi leur communication. Les correspondants administratifs (gestionnaire/agent) sont représentés par des processus qui échangent des informations à travers un protocole (agent/gestionnaire).

L'échange entre agent (administré) et gestionnaire (administrateur) se fait:

- Soit par un mécanisme de demande du gestionnaire/réponse de l'agent,
- soit par un compte rendu spontané de l'administré à l'administrateur à la suite d'un événement.

Nous obtenons ainsi le modèle de la gestion système (figure 1).

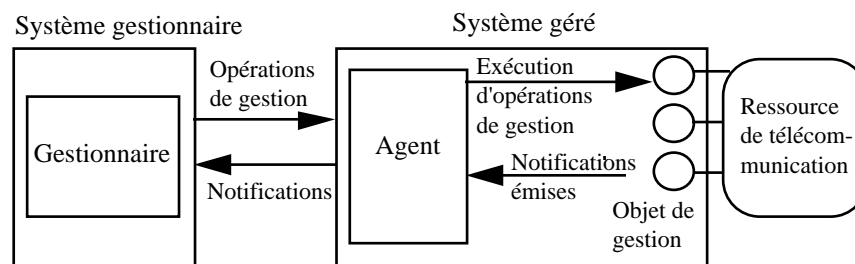


Figure 1 : Interactions gestionnaire-agent-objets gérés

2 Les services CMISE

La gestion système est une application; elle se localise donc en couche 7 du modèle de référence OSI. Les flux applicatifs générés par cette application s'appuieront comme tous les flux sur les services rendus par les couches inférieures et les autres services de la couche 7. Parmi ces derniers, on recense ACSE (Association Control Service Element) qui gère les associations d'application entre deux processus de gestion (établissement, contrôle, libération), ROSE (Remote Operation Service Element) qui permet l'invocation d'opérations distantes et CMISE (Common Management Information Service Element) qui fournit les services de gestion pour les applications de gestion. C'est le protocole CMIP (Common Management Information Protocol) qui fournit le support nécessaire à la communication entre entités d'applications utilisatrices du service CMIS.

La gestion système manipule des objets sur lesquels elle effectue des opérations et des notifications. Les opérations sont celles habituelles des bases de données, soit la création, la suppression, la mise à jour et la lecture d'une valeur auxquelles s'ajoute l'exécution d'action sur l'objet associé. La notification est de type rapport d'événements.

CMISE offre les moyens d'effectuer des échanges entre processus de gestion pour réaliser ces opérations et notifications. Les échanges sont de type demande et demande-réponse.

CMISE assure la mise en oeuvre des services suivants:

- M-CREATE qui permet à un gestionnaire de créer un objet dans la MIB de l'agent,
- M-DELETE qui permet à un gestionnaire de supprimer un objet dans la MIB de l'agent,
- M-ACTION, confirmée ou non, qui permet à un gestionnaire de demander qu'une action soit effectuée par un objet géré,
- M-EVENT-REPORT, confirmée ou non, qui permet à un agent de transmettre la notification émise par un objet géré à un gestionnaire,
- M-GET qui permet à un gestionnaire de lire les valeurs d'attributs d'objets gérés,
- M-SET, confirmée ou non, qui permet à un gestionnaire de modifier les valeurs d'attributs,
- M-CANCEL-GET qui permet d'annuler un service M-GET invoqué.

M-CREATE, M-DELETE, M-GET et M-CANCEL-GET sont des services opérant toujours en mode confirmé.

CMISE permet de demander un ensemble d'objets gérés à travers une demande de service grâce aux mécanismes de cadrage et de filtrage. Le cadrage (scoping) permet de sélectionner un sous-arbre dans la hiérarchie de nommage en spécifiant la racine et la profondeur de ce sous-arbre. La demande de service s'applique ensuite à tous les objets du domaine ainsi délimité.

L'objet géré de base (base object) est défini comme la racine de l'arbre de nommage à partir duquel le cadrage doit être appliqué. Quatre spécifications de niveau de cadrage sont définies indiquant si le filtre doit être appliqué :

- à l'objet de base seul (base object only),
- aux subordonnés de niveau N de l'objet de base (Nth level subordinates only),
- à l'objet de base et à tous ses subordonnés jusqu'au niveau N, ce niveau compris (base object and all subordinates down to and including the Nth level),
- à l'objet de base et à tous ses subordonnés (ensemble du sous-arbre) (entire subtree).

Lorsque le paramètre de cadrage n'est pas spécifié, l'objet géré de base est le seul objet sélectionné.

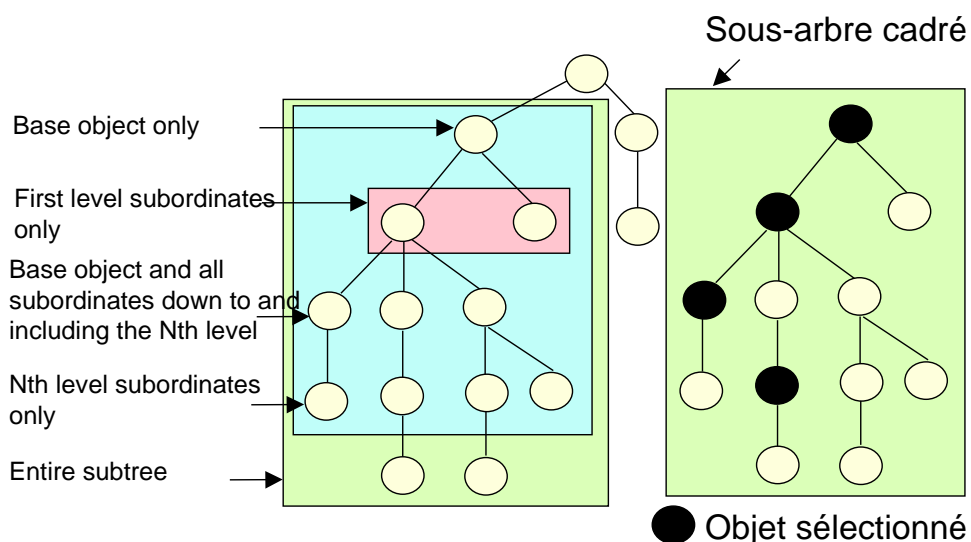


Figure 2: Cadrage et filtrage

Le filtrage (Filtering) consiste à effectuer sélectivement l'opération de gestion sur les objets qui sont dans le domaine de la demande en appliquant un critère de filtrage aux attributs de l'objet.

Un filtre est un ensemble de une ou plusieurs assertions sur la présence ou les valeurs des attributs d'un objet géré sélectionné par le mécanisme de cadrage. Si le filtre comprend

plusieurs assertions, elle sont regroupées à l'aide d'opérateurs logiques ET (AND), OU (OR) et NON (NOT). Si le résultat de l'application du filtre à un objet géré est positif, cet objet est retenu pour l'exécution de l'opération. Lorsque le paramètre de filtrage n'est pas spécifié, tous les objets gérés cadrés sont sélectionnés.

Les assertions sont :

- equality (égalité) : la valeur d'un attribut fait l'objet de tests d'égalité par rapport à une valeur donnée. Ex : operationalState = 'disabled',
- greaterOrEqual (supérieur ou égal) : la valeur d'un attribut est testée en fonction d'une valeur donnée, afin de déterminer si elle est plus grande. Ex : bitrate greaterOrEqual 1.000.000,
- lessOrEqual (inférieur ou égal) : la valeur d'un attribut est comparée à une valeur donnée pour déterminer si elle est plus petite. Ex : bitrate lessOrEqual 10.000.000,
- present (l'attribut est présent) : Ex : present (administrativeState),
- substrings (la valeur d'attribut comprend les sous-chaînes dans l'ordre indiqué) [crossconnectionId substrings ("xcon", "xc")]
- subset of (Tous les membres indiqués sont présents dans la liste de valeurs de l'attribut précisé) [crossConnectionList subset of (xcon1, xcon2, xcon3)]
- superset of (Tous les membres de la liste de valeur de l'attribut sont présents dans la liste indiquée) [crossConnectionList superset of (xcon1, xcon2, xcon3, xcon4, xcon5)]
- non-null-intersection : (au moins un des membres indiqués est présent dans la valeur d'attribut) [crossConnectionList non-null-intersection (xcon1, xcon2)]

On peut noter les similarités entre cadrage (scoping) et filtrage (filtering), et la requête SQL SELECT dans le monde des bases de données relationnelles. L'opération ainsi cadrée peut être effectuée en mode atomique (atomic) ou au mieux (best effort). Il s'agit là de la fonctionnalité de synchronisation. Le mode "atomique" implique que l'opération sera exécutée soit correctement sur tous les objets, soit sur aucun. Le mode "au mieux" stipule que l'opération n'est effectuée que sur les objets qui admettent l'opération.

Illustrons rapidement les rôles respectifs de ACSE, CMISE et ROSE sur un exemple (figure 3). Considérons qu'un gestionnaire désire connaître auprès d'un agent l'état opérationnel d'un commutateur ATM. Il s'agira donc de lire la valeur de l'attribut état opérationnel de l'objet commutateur dans la MIB de l'agent. Le gestionnaire commence par établir une association avec l'agent en invoquant le service offert par ACSE. Une fois l'association établie, le gestionnaire utilise le service offert par CMISE afin d'émettre une requête de lecture de l'objet et de recevoir la réponse (5, 8). ROSE, dont les services sont utilisés par CMISE (6,7), supporte l'émission de la requête et l'envoi de la réponse.

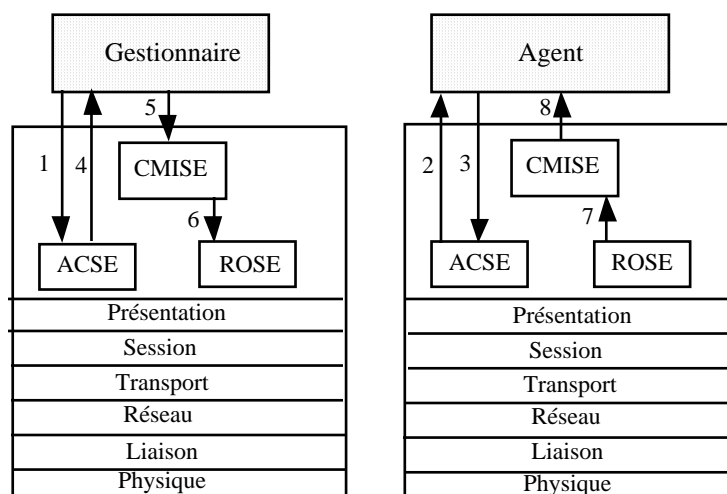


Figure 3 : Exemple de communication de gestion

Dans les paragraphes qui suivent sont décrits les paramètres de chacun des services CMISE à travers des tableaux. Dans ces derniers, M signifie Mandatory (obligatoire), C est mis pour Conditional, U pour User (à la diligence de l'utilisateur) et - signifie l'absence du paramètre correspondant. Lorsque le symbole = est présent dans un champ de réponse, il indique que la valeur du champ est identique à celle du champ correspondant dans la requête.

Les paramètres du service M-EVENT-REPORT sont l'identifiant de la requête (Invoke Identifier), le mode du service M-EVENT-REPORT (confirmé ou non confirmé), le nom de la classe d'objet (Managed Object Class) dont une des instances (Managed Object Instance) a émis la notification, le type d'événement émis qui indique le nom de la notification (Event Type). Parmi les paramètres facultatifs, figurent l'instant d'émission de la notification (Event Time), et des informations associées à l'événement (Event Information).

Si le mode est confirmé, alors le gestionnaire émet une réponse en direction de l'agent qui contient le même identifiant de requête. Cette réponse peut rappeler certains paramètres, et contient un type d'erreur si une erreur est survenue (e.g., no such event) (tableau 1).

Nom du paramètre	Requête	Réponse
Invoke Identifier	M	M(=)
Mode	M	-
Managed Object Class	M	U
Managed Object Instance	M	U
Event Time	U	-
Event Type	M	C(=)
Event Information	U	-
Current Time	-	U
Event Reply	-	C
Errors	-	C

Tableau 1: Paramètres de la primitive M-EVENT-REPORT

La primitive M-GET fonctionne toujours en mode confirmé (requête/réponse). L'opération M-GET tout comme les opérations M-SET, M-ACTION et M-DELETE utilisent les principes de cadrage et filtrage.

Le cadrage comprend l'identification du ou des objets gérés auxquels un filtre doit être appliqué.

Le filtrage comprend l'application d'un ensemble de tests à chaque élément de l'ensemble des objets gérés sélectionnés par le cadrage, pour en extraire un sous-ensemble.

Le sous-ensemble des objets sélectionnés par le cadrage qui satisfont le filtre est retenu pour l'opération.

A l'émission, le paramètre Base Object Class identifie la classe dont une des instances (Base Object instance) sera l'objet racine du sous-arbre de nommage sélectionné par l'opération de cadrage. Le paramètre cadre (Scope) indique la profondeur du sous-arbre avec les possibilités suivantes : Base object only, first level subordinates only, Nth level subordinate only, entire subtree. Le paramètre filtre (Filter) est appliqué au sous arbre sélectionné par le cadrage.

Le paramètre contrôle d'accès (Access Control) peut être utilisé pour ne permettre que les accès autorisés aux objets de gestion. La synchronisation peut être soit atomique soit au mieux. La liste d'identifiants d'attributs (Attribute Identifier List) contient l'identifiant de chaque attribut dont la valeur est souhaitée par le gestionnaire.

A une requête M-GET peuvent correspondre plusieurs réponses (réponse multiple), en fonction du nombre d'objets sélectionnés par les opérations de cadrage et de filtrage (une réponse M-GET par objet sélectionné). Chaque réponse possède un identifiant de réponse (Linked identifier), le nom de la classe d'objet (Managed Object Class) dont l'instance a été sélectionnée (Managed Object Instance), l'instant de génération de la réponse (Current

Time), et la liste d'attributs avec leurs valeurs associées (Attribute list). Une erreur peut éventuellement survenir; dans ce cas un identifiant d'erreur est rapporté au gestionnaire (e.g., no such object class, no such object instance, etc.) (tableau 2).

Nom du paramètre	Requête	Réponse
Invoke Identifier	M	M
Linked Identifier	-	C
Base Object Class	M	-
Base Object Instance	M	-
Scope	U	-
Filter	U	-
Access Control	U	-
Synchronisation	U	-
Attribute Identifier List	U	-
Managed Object Class	-	C
Managed Object Instance	-	C
Current Time	-	U
Attribute List	-	C
Errors	-	C

Tableau 2 : Paramètres de la primitive M-GET

Les paramètres de la primitive M-SET sont identiques à ceux de la requête M-GET à l'exception de l'attribut Modification List (tableau 3). Chaque élément de cette liste contient un identifiant d'attribut, un opérateur de modification (replace, add value, remove value, set to default) et de manière optionnelle une ou des valeurs d'attributs. Add value et remove value s'appliquent à des attributs dont les valeurs sont des listes.

Nom du paramètre	Requête	Réponse
Invoke Identifier	M	M
Linked Identifier	-	C
Mode	M	-
Base Object Class	M	-
Base Object Instance	M	-
Scope	U	-
Filter	U	-
Access Control	U	-
Synchronisation	U	-
Managed Object Class	-	C
Managed Object Instance	-	C
Modification List	M	U
Current Time	-	C
Attribute List	-	C
Errors	-	C

Tableau 3 : Paramètres de la primitive M-SET

M-ACTION ressemble aux requêtes M-GET et M-SET, à la différence qu'elle ne transporte pas des attributs mais des actions (tableau 4). Les informations associées à l'action sont

Action Type qui correspond au type d'action tel qu'il est défini par le formulaire ACTION, Action Information indiquant l'information nécessaire à l'invocation de l'action.

Nom du paramètre	Requête	Réponse
Invoke Identifier	M	M
Linked Identifier	-	C
Mode	M	-
Base Object Class	M	-
Base Object Instance	M	-
Scope	U	-
Filter	U	-
Access Control	U	-
Synchronisation	U	-
Managed Object Class	-	C
Managed Object Instance	-	C
Action Type	M	C(=)
Action Information	-	C
Action Reply	-	C
Current Time	-	U
Errors	-	C

Tableau 4 : Paramètres de la primitive M-ACTION

Les paramètres de la primitive M-CREATE sont le nom de la classe d'objet de gestion (Managed Object Class) dont une instance sera créée (Managed Object Instance). Le paramètre Superior Object Instance identifie l'objet dans l'arbre de nommage auquel l'instance qui sera créée sera rattachée. Attribute list spécifie les valeurs d'attributs pour la nouvelle instance. Il est aussi possible d'indiquer un paramètre Reference Object Instance qui identifie une instance de la classe dont les valeurs d'attributs seront les valeurs par défaut des attributs de la nouvelle instance (tableau 5).

Nom du paramètre	Requête	Réponse
Invoke Identifier	M	M(=)
Managed Object Class	M	U
Managed Object Instance	U	C
Superior Object instance	U	-
Access Control	U	-
Reference Object Instance	U	-
Attribute List	U	C
Current Time	-	U
Errors	-	C

Tableau 5: Paramètres de la primitive M-CREATE

A la différence de M-CREATE, M-DELETE supporte le cadrage et le filtrage. En effet, il est possible de supprimer un ensemble d'objets à travers une unique invocation du service M-DELETE (tableau 6).

Nom du paramètre	Requête	Réponse
Invoke Identifier	M	M
Linked Identifier	-	C
Base Object Class	M	-
Base Object Instance	M	-
Scope	U	-
Filter	U	-
Access Control	U	-
Synchronisation	U	-
Managed Object Class	-	C
Managed Object Instance	-	C
Current Time	-	U
Attribute List	-	C
Errors	-	C

Tableau 6: Paramètres de la primitive M-DELETE

Le paramètre important de la primitive M-CANCEL-GET est l'identifiant de la requête M-GET à annuler (Get Invoke Identifier) (tableau 7).

Nom du paramètre	Requête	Réponse
Invoke Identifier	M	M(=)
Get Invoke Identifier	M	-
Errors	-	C

Tableau 7: Paramètres de la primitive M-CANCEL-GET

Etant donné la complexité de CMIS, la plupart des plateformes de gestion de réseau TMN offrent des interfaces de programmation de plus haut niveau [Miller 94].

3 Le protocole CMIP

Le protocole CMIP est exprimé par un ensemble d'opérations CMIP correspondant aux services CMISE. Si un service peut fonctionner dans les modes confirmé et non-confirmé, deux opérations sont alors utilisées, une pour chaque mode. C'est le cas du service M-EVENT-REPORT pour lequel deux opérations sont définies à savoir : m-EventReport et m-EventReport-Confirmed. C'est aussi le cas pour M-SET (m-Set et m-Set-Confirmed), M-ACTION (m-Action et m-Action-Confirmed). Si le service est toujours confirmé, il est traduit en une seule opération : M-GET (m-Get), M-CREATE (m-Create), M-DELETE (m-Delete), M-CANCEL-GET (m-CancelGet). Enfin, pour transporter des réponses multiples, une opération m-Linked-Reply est utilisée. Ce sont donc 11 opérations CMIP qui sont définies (tableau 8).

m-EventReport
m-EventReport-Confirmed
m-Linked-Reply
m-Get
m-Set
m-Set-Confirmed
m-Action
m-Action-Confirmed

m-Create
m-Delete
m-CancelGet

Tableau 8 : Opérations CMIP

4 Exemple d'application de CMIS

Afin d'illustrer le fonctionnement de CMIS, considérons l'arbre de nommage suivant (Figure 4):

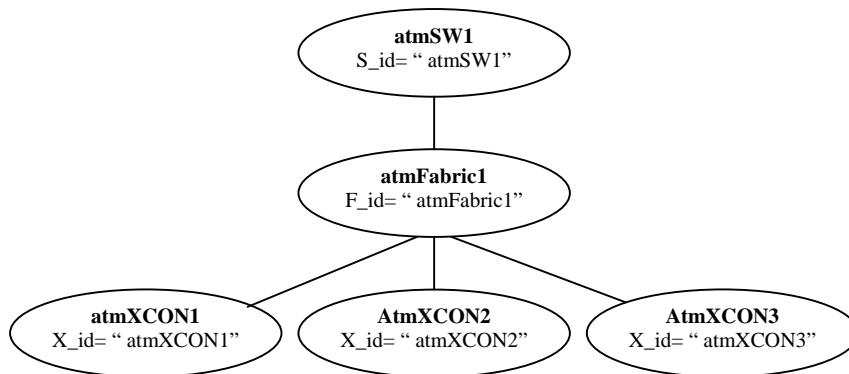


Figure 4 : Exemple d'arbre de nommage

Décrivons l'envoi à un gestionnaire d'un événement émis par l'objet atmXCON1 (Figure 5). Le nom de l'événement est stateChange indiquant que l'état opérationnel de l'objet XCON1 a désormais pour valeur « disabled ». Le mode choisi est non confirmé.

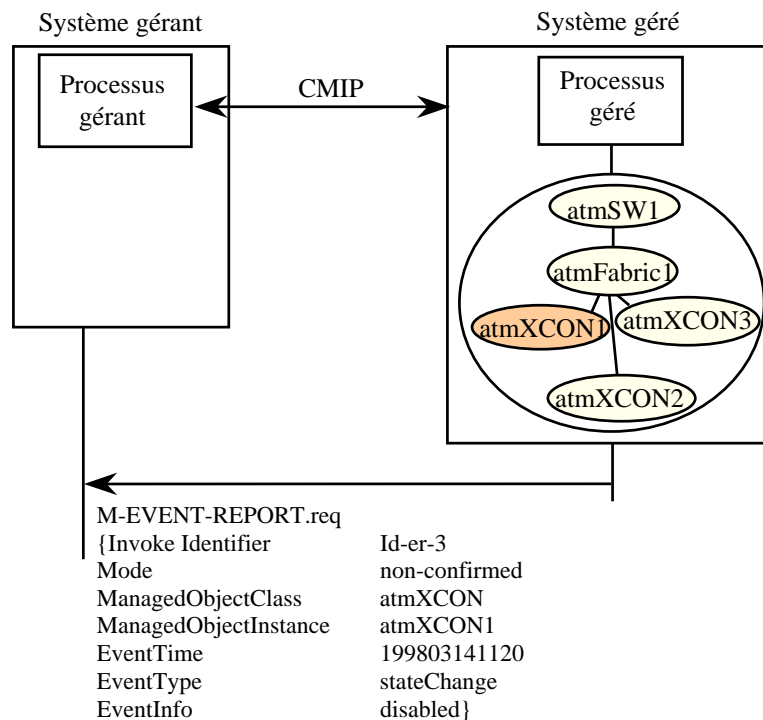


Figure 5 : Exemple de requête M-EVENT-REPORT non confirmée

Décrivons les requêtes de lecture des attributs `crossConnectionId` et `administrativeState` de l'ensemble des connexions dans le commutateur `atmSW1` ainsi que des réponses retournées par chacune des connexions (Figure 6).

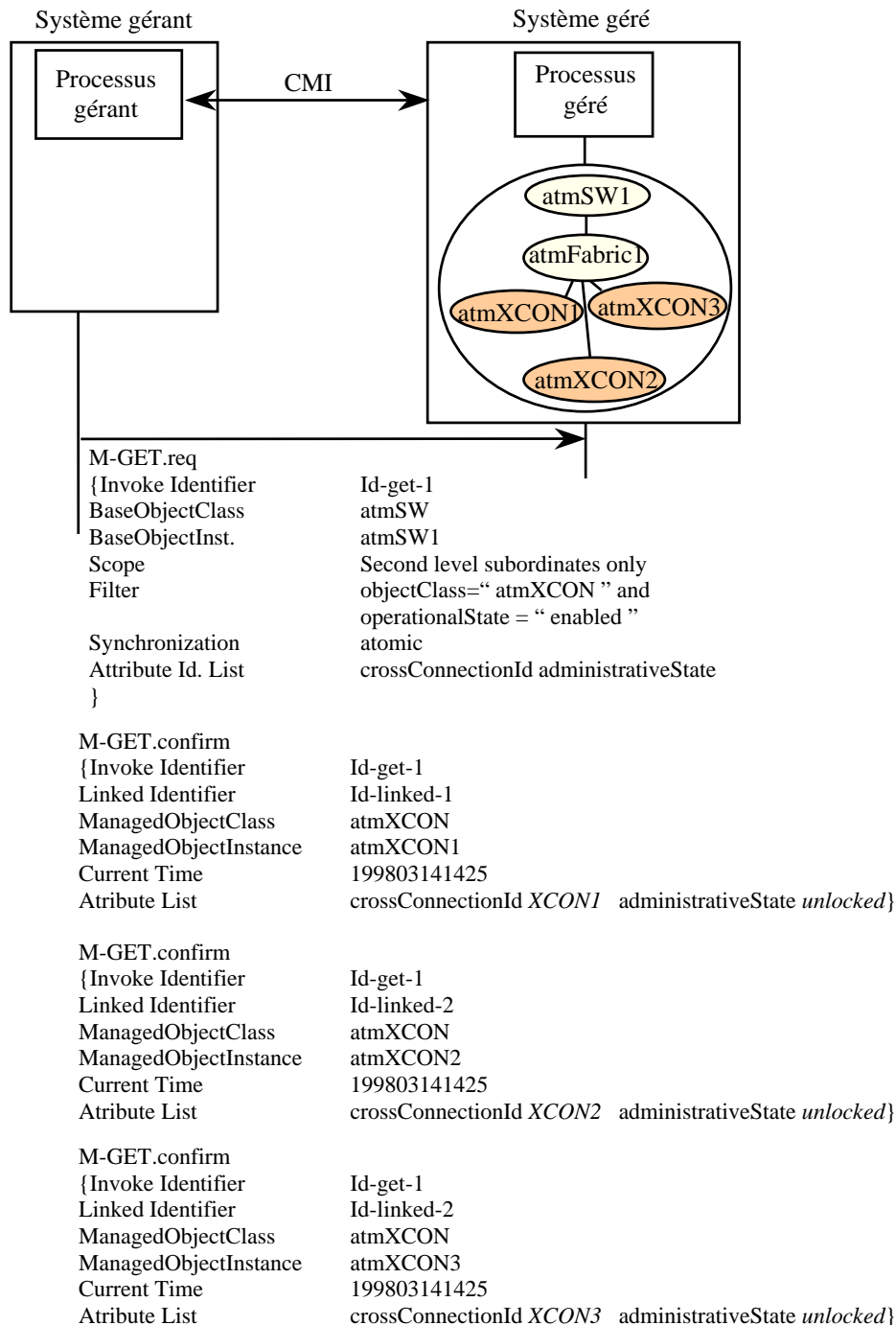


Figure 6 : Exemple de de requête M-GET et des réponses associées

Décrivons les requêtes de modification de l'attribut `administrativeState` de la connexions `atmXCON1` dans le commutateur `atmSW1` ainsi que la confirmation émise par l'agent du commutateur (Figure 7).

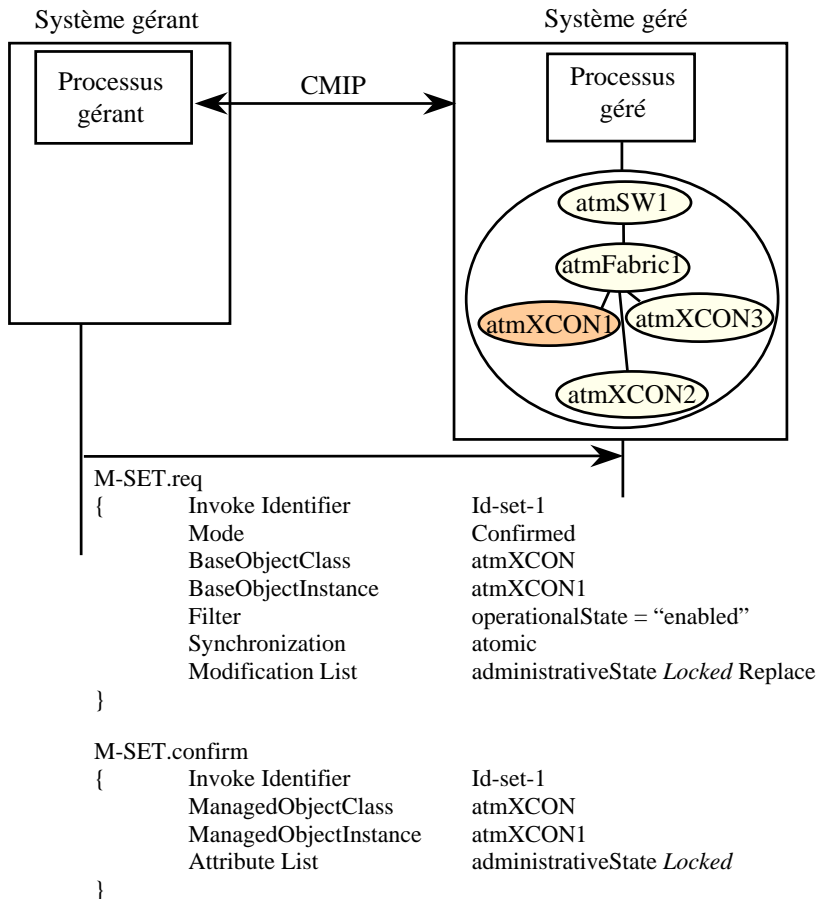


Figure 7 : Exemple de requête M-SET et de la réponse associée

La requête suivante (Figure 8) transporte une action « reset » destinée à l'objet atmFabric1 dont le but est de réinitialiser ce fabric (en français réseau de connexion).

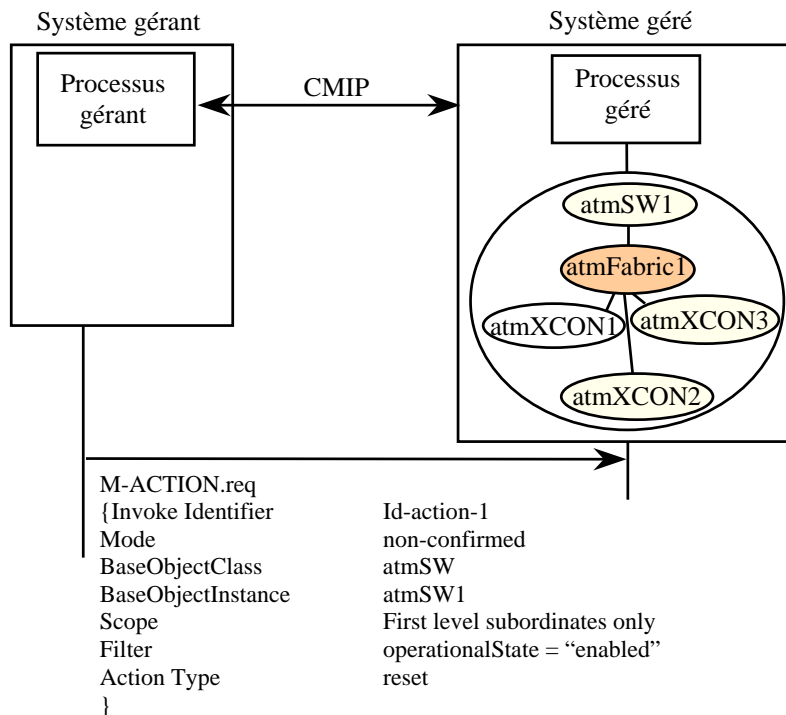


Figure 8 : Exemple de requête M-ACTION non confirmée

Décrivons la requête de suppression de l'ensemble des connexions du commutateur atmSW1 ainsi que les réponses qui confirment les suppressions de ces connexions (Figure 9).

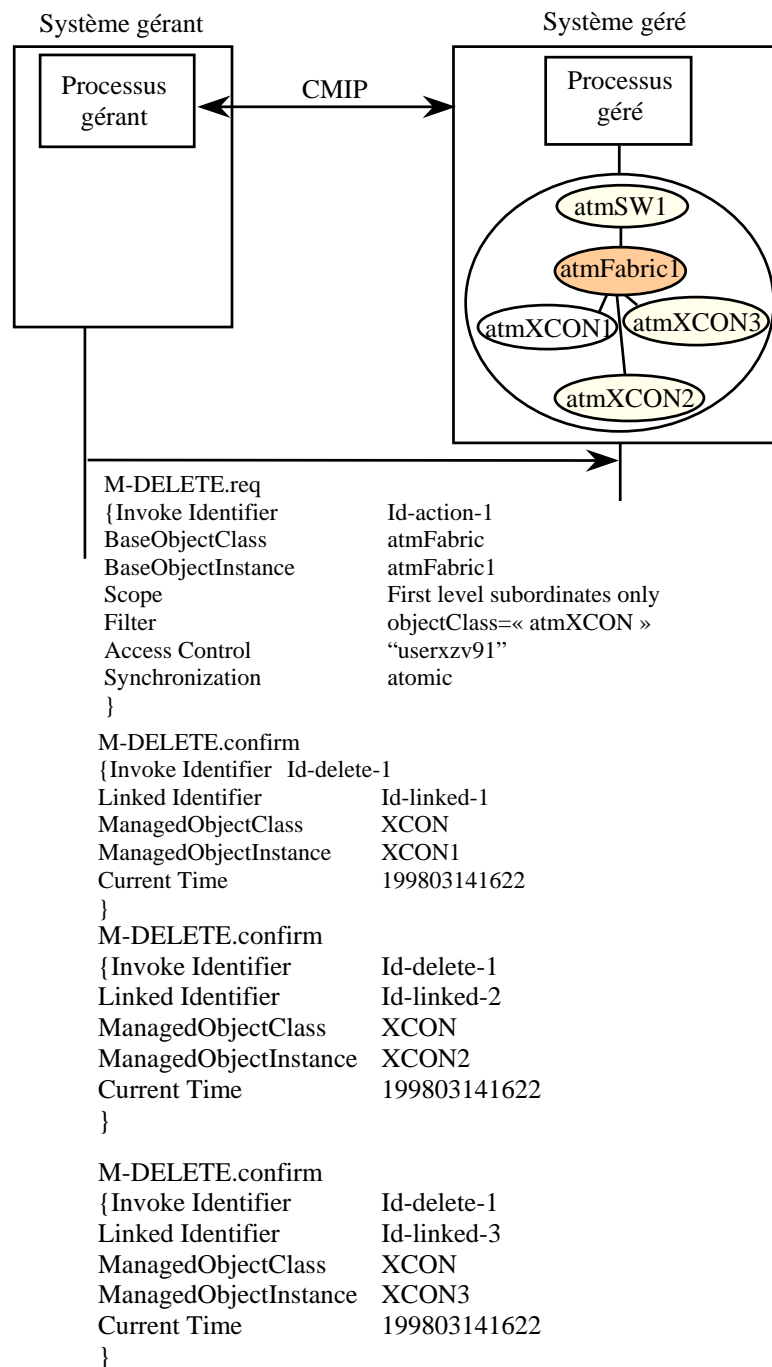


Figure 9 : Exemple de requête M-DELETE et des réponses associées

La requête suivante décrit l'annulation d'une requête M-GET d'invoke identifiant id-get-1 (Figure 10). On suppose qu'une telle requête M-GET n'existe pas.

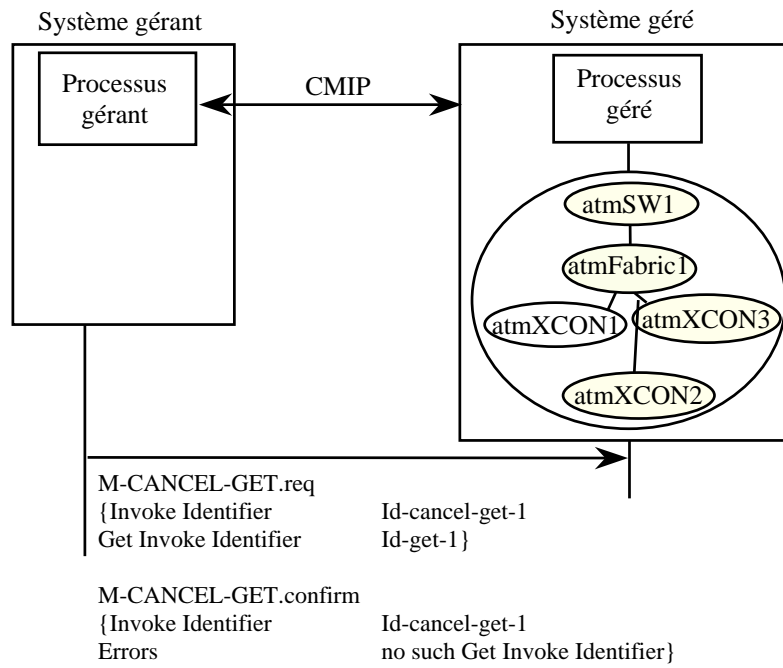


Figure 10 : Exemple de requête M-CANCEL-GET et de réponse associée