

Groupe de travail Réseau
Request for Comments : 3060
 Catégorie : En cours de normalisation

Traduction Claude Brière de L'Isle

B. Moore, IBM
 E. Ellesson, LongBoard, Inc.
 J. Strassner & A. Westerinen
 Cisco Systems
 février 2001

Spécification du modèle d'information de cœur de politique - version 1

Statut de ce mémoire

Le présent document spécifie un protocole Internet en cours de normalisation pour la communauté de l'Internet, et appelle à des discussions et des suggestions pour son amélioration. Prière de se reporter à l'édition actuelle du STD 1 "Normes des protocoles officiels de l'Internet" pour connaître l'état de normalisation et le statut de ce protocole. La distribution du présent mémoire n'est soumise à aucune restriction.

Notice de copyright

Copyright (C) The Internet Society (2001). Tous droits réservés.

Résumé

Ce document présente le modèle d'informations en mode objet pour la représentation des informations de politique développé conjointement par le groupe de travail Cadre de politique de l'IETF et comme des extensions à l'activité de modèle d'informations communes (CIM, *Common Information Model*) dans l'équipe sur la gestion répartie (DTMF, *Distributed Management Task Force*). Ce modèle définit deux hiérarchies de classes d'objet : les classes structurelles qui représentent des informations de politique et des commandes de politiques, et les classes d'association qui indiquent comment les instances des classes structurelles sont en rapport les unes avec les autres. Des documents ultérieurs définiront les transpositions de ce modèle d'information en diverses mises en œuvre concrètes, par exemple, dans un répertoire qui utilise LDAPv3 comme protocole d'accès.

Table des Matières

1. Introduction.....	2
2. Politiques de modélisation.....	3
2.1 Portée de politique.....	4
2.2 Modèle déclaratif ou procédural.....	4
3. Vue d'ensemble du modèle d'information de cœur de politique.....	5
4. Hiérarchies héritées pour le modèle d'information de cœur de politique.....	7
4.1 Implications de l'héritage CIM.....	8
5. Détails du modèle.....	8
5.1 Conditions et actions réutilisables ou spécifiques de règles.....	9
5.2 Rôles.....	9
5.3 Heure locale et UTC dans PolicyTimePeriodConditions.....	12
5.4 Types de données CIM.....	13
5.5 Comparaison entre spécifications de classe CIM et LDAP.....	13
6. Définitions de classe.....	14
6.1 Classe abstraite "Policy".....	14
6.2 Classe "PolicyGroup".....	15
6.3 Classe "PolicyRule".....	16
6.4 Classe abstraite "PolicyCondition".....	19
6.5 Classe "PolicyTimePeriodCondition".....	20
6.6 Classe "VendorPolicyCondition".....	24
6.7 Classe abstraite "PolicyAction".....	25
6.8 Classe "VendorPolicyAction".....	25
6.9 Classe "PolicyRepository".....	26
7. Définitions d'association et d'agrégation.....	26
7.1 Associations.....	27
7.2 Agrégations.....	27
7.3 Agrégation abstraite "PolicyComponent".....	27
7.4 Agrégation "PolicyGroupInPolicyGroup".....	27
7.5 Agrégation "PolicyRuleInPolicyGroup".....	28
7.6 Agrégation "PolicyConditionInPolicyRule".....	28
7.7 Agrégation "PolicyRuleValidityPeriod".....	29

7.8 Agrégation "PolicyActionInPolicyRule".....	30
7.9 Association abstraite "PolicyInSystem".....	31
7.10. Association faible "PolicyGroupInSystem".....	32
7.11 Association faible "PolicyRuleInSystem".....	32
7.12 Association "PolicyConditionInPolicyRepository".....	33
7.13 Association "PolicyActionInPolicyRepository".....	33
7.14 Agrégation "PolicyRepositoryInPolicyRepository".....	34
8. Propriété intellectuelle.....	34
9. Remerciements.....	34
10. Considérations pour la sécurité.....	35
11. Références.....	36
12. Adresse des auteurs.....	36
13 Appendice A Identification de classe dans une mise en œuvre CIM native.....	37
13.1 Instances de désignation de PolicyGroup et PolicyRule.....	37
13.2 Instances de dénomination de PolicyCondition et ses sous classes.....	38
13.3 Instances de désignation de PolicyAction et ses sous classes.....	40
13.4 Instances de désignation de PolicyRepository.....	41
13.5 Rôle de la propriété CreationClassName pour la désignation.....	41
13.6 Références d'objet.....	42
14. Appendice B : Format d'objet géré de cœur de politique.....	42
15. Déclaration complète de droits de reproduction.....	55

1. Introduction

Le présent document présente le modèle d'informations en mode objet pour représenter les informations de politique qui sont en cours de développement conjoint par le groupe de travail Cadre de politique de l'IETF et comme une activité d'extension au modèle d'informations communes (CIM, *Common Information Model*) dans l'équipe de gestion répartie (DTMF, *Distributed Management Task Force*). Ce modèle définit deux hiérarchies de classes d'objet : les classes structurelles qui représentent des informations de politique et des commandes de politiques, et des classes d'association qui indiquent comment des instances des classes structurelles se rapportent les unes aux autres. Des documents ultérieurs définiront les transpositions de ce modèle d'informations en diverses mises en œuvre concrètes, par exemple, en un répertoire qui utilise LDAPv3 comme protocole d'accès. Les composants du schéma CIM sont disponibles via l'URL : <http://www.dmtf.org/spec/cims.html> [1].

Les classes et associations de politique définies dans ce modèle sont suffisamment génériques pour leur permettre de représenter toutes sortes de politiques. Cependant, on s'attend à ce que leur application initiale dans l'IETF soit pour représenter des politiques en rapport avec la qualité de service (DiffServ et IntServ) et avec IPSec. Les modèles de politique pour des domaines spécifiques de l'application tels que ceux-là peuvent étendre le modèle cœur de diverses façon. La façon préférée est d'utiliser directement les classes PolicyGroup, PolicyRule, et PolicyTimePeriodCondition, comme fondement de la représentation de la communication des informations de politique. Ensuite, des sous classes spécifiques déduites de PolicyCondition et de PolicyAction peuvent capturer des définitions spécifiques de l'application des conditions et actions des politiques.

Deux sous classes, VendorPolicyCondition et VendorPolicyAction, sont aussi incluses dans ce document, pour fournir un mécanisme d'extension standard aux extensions spécifiques de fabricant au modèle d'informations de cœur de politique.

Le présent document entre dans un cadre général de représentation, déploiement, et gestion des politiques qui est développé par le groupe de travail PolicyFramework. Il trouve son origine dans un travail qui était fait au départ pour la spécification de réseaux à capacité de répertoire (DEN, *Directory-enabled Networks*) [5]. Le travail sur la spécification DEN par le groupe de travail ad hoc DEN lui-même a été achevé. D'autres travaux pour normaliser les modèles qui y étaient contenus seront de la responsabilité des groupes de travail de l'effort CIM dans l'équipe Gestion répartie (DTMF, *Distributed Management Task Force*). La normalisation DMTF du modèle de cœur de politique est de la responsabilité du groupe de travail Politique SLA dans DMTF.

Le présent document est organisé de la manière suivante :

- o La Section 2 donne une vue générale d'ensemble des politiques et de la façon de les modéliser.
- o La Section 3 présente un survol des classes et associations composant le modèle d'informations de cœur de politique.
- o Le reste de ce document présente la spécifications détaillée de chacune des classes et associations.
- o L'Appendice A résume le mécanisme de désignation des mises en œuvres CIM natives. D'autres transpositions, comme LDAPv3, auront leur propre mécanisme de désignation.
- o L'Appendice B reproduit la spécification du format d'objet géré (MOF, *Managed Object Format*) de cœur de politique de DMTF.

Dans le présent document, les mots clés "DOIT", "NE DOIT PAS", "EXIGE", "DEVRA", "NE DEVRA PAS", "DEVRAIT", "NE DEVRAIT PAS", "RECOMMANDE", "PEUT", et "FACULTATIF" sont à interpréter comme décrit dans le BCP 14, [RFC2119].

2. Politiques de modélisation

Les classes qui comportent le modèle d'informations de cœur de politique sont destinées à servir de hiérarchie de classe extensible (par spécialisation) pour définir les objets de politique qui permettent aux développeurs d'application, aux administrateurs de réseau, et aux administrateurs de politiques de représenter des politiques de différents types.

Une façon de penser un réseau contrôlé par une politique est de modéliser d'abord le réseau comme un automate à états et ensuite d'utiliser la politique pour contrôler dans quel état devrait être, ou est autorisé à être, un appareil contrôlé par la politique, à tout instant. Selon cette approche, la politique est appliquée en utilisant un ensemble de règles de politique. Chaque règle de politique consiste en un ensemble de conditions et en un ensemble d'actions. Les règles de politique peuvent être agrégées en des groupes de politiques. Ces groupes peuvent être incorporés, pour représenter une hiérarchie de politiques.

L'ensemble des conditions associées à une règle de politique spécifique quand la règle de politique est applicable. L'ensemble de conditions peut être exprimé soit comme un ensemble OUIé d'ensembles ETixés de déclarations de conditions, soit comme un ensemble ETixé d'ensembles OUIés de déclarations. Les déclarations individuelles de condition peuvent aussi être niées. Ces combinaisons sont appelées, respectivement, forme normale disjonctive (DNF, *Disjunctive Normal Form*) et forme normale conjonctive (CNF, *Conjunctive Normal Form*) pour les conditions.

Si l'ensemble de conditions associées à une règle de politique s'évalue à VRAI, un ensemble d'actions qui, soit conserve l'état en cours de l'objet, soit fait passer l'objet à un nouvel état, peut être exécuté.

Pour l'ensemble d'actions associées à une règle de politique, il est possible de spécifier un ordre d'exécution, ainsi qu'une indication de si l'ordre est exigé ou simplement recommandé. Il est aussi possible d'indiquer que l'ordre dans lequel les actions sont exécutées n'a pas d'importance.

Les règles de politique elles-mêmes peuvent recevoir une priorité. Une raison courante pour le faire est d'exprimer une politique globale qui a un cas général avec quelques exceptions spécifiques.

Par exemple, une règle de politique de qualité de service générale peut spécifier que le trafic originaire des membres du groupe d'ingénierie va obtenir le service Bronze. Une seconde règle de politique pourrait exprimer une exception : le trafic généré par John, un membre spécifique du groupe d'ingénierie, va obtenir le service Gold. Comme le trafic originaire de John satisfait les conditions des deux règles de politique, et comme les actions associées aux deux règles sont incompatibles, une priorité doit être établie. En donnant à la seconde règle (l'exception) une plus forte priorité qu'à la première règle (le cas général) un administrateur de politique peut obtenir l'effet désiré : le trafic généré par John obtient le service Gold, et le trafic originaire de tous les autres membres du groupe d'ingénierie obtient le service Bronze.

Les politiques peuvent être utilisées de façon autonome ou agrégées en groupes de politiques pour effectuer des fonctions plus élaborées. Les politiques autonomes sont appelées des règles de politique. Les groupes de politiques sont des agrégations de règles de politique, ou agrégations de groupes de politiques, mais pas les deux. Les groupes de politiques peuvent modéliser des interactions entrelacées entre les objets qui ont des interdépendances complexes. Par exemple, une politique sophistiquée de connexion d'utilisateur qui établit des accès aux applications, à la sécurité, et reconfigure les connexions réseau sur la base d'une combinaison de l'identité de l'utilisateur, de la localisation du réseau, de la méthode de connexion et de l'heure. Un groupe de politiques représente une unité de réutilisation et de capacité de gestion en ce que sa gestion est traitée par un groupe identifiable d'administrateurs et ses règles de politique seront appliquées de façon cohérente.

Les politiques autonomes sont celles qui peuvent être exprimées dans une déclaration simple. Elles peuvent être représentées effectivement dans des schémas ou des bases de données d'informations de gestion (MIB, *Management Information Base*). Des exemples en sont des allocations de VLAN, de simples demandes de QS OUI/NON, et des allocations d'adresse IP. Un objectif spécifique de conception de ce modèle est de prendre en charge les politiques autonomes aussi bien qu'agrégées.

Les groupes et règles de politique peuvent être classés par leur objet et leur intention. Cette classification est utile pour interroger ou grouper les règles de politique. Cela indique si la politique est utilisée pour motiver quand ou comment survient une action, ou pour caractériser des services (qui peuvent être utilisés, par exemple, pour lier des clients aux

services réseau). On décrit chacun de ces concepts plus en détail.

- o Les politiques de motivation ne visent qu'à établir si et comment un objectif de politique est accompli. Les politiques de configuration et d'usage sont des genres spécifiques de politiques de motivation. Un autre exemple est la programmation des sauvegardes de fichier sur la base d'une activité d'écriture sur disque de 8 h à 15 h, M-F.
- o Les politiques de configuration définissent l'établissement par défaut (ou générique) d'une entité gérée (par exemple, un service réseau). Des exemples de politiques de configuration sont l'établissement d'un service de transmission réseau ou d'une file d'attente d'impression hébergée par le réseau.
- o Les politiques d'installation définissent ce qui peut ou ne peut pas être mis sur un système ou un composant, ainsi que la configuration des mécanismes qui effectuent l'installation. Les politiques d'installation représentent normalement des permissions administratives spécifiques, et peuvent aussi représenter des relations d'interdépendance entre différents composants (par exemple, pour achever l'installation du composant A, les composants B et C doivent avoir été préalablement bien installés ou désinstallés).
- o Politiques d'erreur et d'événement. Par exemple, si un appareil a une défaillance entre 8 h et 21 h, appeler l'administrateur système, autrement, appeler le bureau d'assistance.
- o Les politiques d'usage contrôlent le choix et la configuration des entités sur la base de données spécifique "d'usage". Les politiques de configuration peuvent être modifiées ou simplement réappliquées par les politiques d'usage. Des exemples de politiques d'usage incluent de mettre à niveau des services de transmission réseau après qu'on a vérifié qu'un usager est membre d'un groupe de service "gold", ou la reconfiguration d'une imprimante pour être capable de traiter la tâche suivante de la file d'attente.
- o Les politiques de sécurité traitent de la vérification que le client est bien ce qu'il prétend être, permettant ou refusant l'accès aux ressources, choisissant et appliquant les mécanismes d'authentification appropriés, et effectuant la comptabilité et l'examen des ressources.
- o Les politiques de service caractérisent les services réseau et autres (ne les utilisent pas). Par exemple, toutes les interfaces de cœur de réseau de large zone devront utiliser un type spécifique de mise en file d'attente. Les politiques de service décrivent les services disponibles dans le réseau. Les politiques d'usage décrivent les liens particuliers d'un client du réseau avec les services disponibles dans le réseau.

Ces catégories sont représentées dans le modèle d'information de cœur de politique par des valeurs spéciales définies pour la propriété PolicyKeywords de la politique de classe abstraite.

2.1 Portée de politique

Les politiques représentent des buts et objectifs d'affaires. Une traduction peut être faite entre ces buts et objectifs et leur réalisation dans le réseau. Un exemple pourrait en être un accord de niveau de service (SLA, *Service Level Agreement*) et ses objectifs et métriques (objectifs de niveau de service (SLO, *Service Level Objectives*) qui sont utilisés pour spécifier les services que le réseau va fournir à un certain client. Le SLA va normalement être écrit dans une terminologie commerciale de haut niveau. Les SLO utilisent des métriques plus spécifiques pour la prise en charge du SLA. Ces descriptions très générales des services et métriques du réseau doivent être traduites en spécifications plus précises, mais aussi indépendantes du fabricant et de l'appareil. Les classes du modèle d'information de cœur de politique sont destinées à servir de fondations à ces spécifications de niveau inférieur, indépendantes du fabricant et de l'appareil.

Il est envisagé que dans le présent document, la définition du modèle d'information de cœur de politique soit de nature générique et soit applicable à la qualité de service (QS), aux applications de réseautage non QS (par exemple, DHCP et IPsec) et aux applications hors réseautage (par exemple, politiques de sauvegarde, accès pour la révision, etc.).

2.2 Modèle déclaratif ou procédural

La conception du modèle d'informations de cœur de politique est influencée par une approche déclarative, non procédurale. Plus formellement, un langage déclaratif est utilisé pour décrire des langages relationnels et fonctionnels. Les langages déclaratifs décrivent les relations entre les variables en termes de fonctions ou règles d'inférence, auxquelles l'interprète ou compilateur peut appliquer un algorithme fixé afin de produire un résultat. Un langage impératif (ou procédural) spécifie une séquence explicite d'étapes à suivre afin de produire un résultat.

Il est important de noter que ce modèle d'informations n'exclut pas l'utilisation des langages procéduraux. Il reconnaît plutôt que les deux langages, déclaratif aussi bien que procédural peuvent être utilisés pour mettre en œuvre la politique. Ce modèle d'informations est mieux vu comme étant déclaratif parce que la séquence des étapes pour faire le traitement de déclarations déclaratives tend à être laissée aux mises en œuvre. Cependant, on a fourni l'option d'exprimer l'ordre désiré des actions d'exécution dans ce modèle d'informations de politique, et pour exprimer si l'ordre est obligatoire ou non. De plus, plutôt que d'essayer de définir les algorithmes ou les ensembles d'instructions ou étapes qui doivent être suivies par une règle de politique, on définit à la place un ensemble de blocs de construction modulaires et de relations qui peuvent être utilisés dans un mode déclaratif ou procédural pour définir les politiques.

Comparons cela à un modèle strictement procédural. En prenant une telle approche on serait obligé de spécifier la séquence de vérification des conditions, et la séquence d'exécution des actions, dans le répertoire même des politiques. Cela constituerait bien sûr une contrainte pour les mises en œuvre. C'est pourquoi le modèle de politique est caractérisé comme déclaratif. C'est-à-dire que le modèle d'informations définit un ensemble d'attributs, et un ensemble d'entités qui contiennent ces attributs. Cependant, il NE définit PAS l'algorithme pour produire un résultat en utilisant les attributs ou une séquence explicite d'étapes pour produire un résultat.

Plusieurs remarques sur la conception et les compromis doivent être mentionnées à cet égard :

1. D'un côté, on voudrait un langage de définition de politique raisonnablement compréhensible par l'homme pour faciliter les définitions et les diagnostics. D'un autre côté, étant donnée la diversité des appareils (en termes de capacités de traitement) qui peuvent agir comme points de décision de politique, on aimerait garder un langage assez simple pour la machine. C'est-à-dire qu'il devrait être relativement simple pour automatiser l'analyse et le traitement du langage dans les éléments de réseau. L'approche retenue est de fournir un ensemble de classes et d'attributs qui puissent être combinés aussi bien dans une approche déclarative que procédurale pour exprimer les politiques qui gèrent les éléments et services du réseau. Le point clé est d'éviter d'essayer de normaliser des règles ou des ensembles d'étapes à suivre pour définir une politique. Cela doit être laissé aux mises en œuvre. L'interopérabilité est réalisée en normalisant les blocs de construction qui sont utilisés pour représenter les données et les informations de politique.
2. Une importante décision à prendre est celle du style de sémantique de la représentation des informations. L'approche déclarative que nous décrivons ne peut pas être un "vrai" modèle déclaratif. Un tel modèle devrait aussi spécifier les algorithmes utilisés pour combiner les informations et règles de politique pour réaliser un comportement particulier. On évite de spécifier les algorithmes pour la même raison qu'on évite de spécifier les ensembles d'étapes à suivre dans une règle de politique. Cependant, la conception du modèle d'informations suit de plus près celle d'un langage déclaratif, et peut être plus facile à comprendre si on utilise un tel modèle conceptuel. Cela conduit à notre troisième point, qui est de reconnaître un manque "d'exhaustivité" et de s'appuyer à la place sur la présentation des informations avec lesquelles va travailler l'entité qui traite la politique.
3. Il est important de contrôler la complexité de la spécification, et de faire un compromis entre la richesse de l'expression des données dans le cœur du modèle d'informations pour faciliter la mise en œuvre et l'utilisation. Il est important de reconnaître le manque collectif d'expérience dans les domaines qui concernent les politiques de contrôle et de gestion des services réseau et donc d'éviter la tentation de viser à "l'exhaustivité". On devrait plutôt s'efforcer de faciliter la définition d'un ensemble de politiques communes qu'exigent le consommateur d'aujourd'hui (par exemple, VPN et QS) et qui permettent des chemins de migration vers la prise en charge de politiques complexes lorsque les consommateurs en auront besoin et que notre compréhension de ces politiques aura évolué grâce à l'expérience. Précisément, dans le contexte du langage de style déclaratif évoqué ci-dessus, il est important d'éviter d'avoir des calculs de prédicat complets comme langage, car cela rendrait insolubles de nombreux problèmes importants comme les vérifications de cohérence et les algorithmes de point de décision de politique. Il est utile de considérer un langage avec des contraintes raisonnables d'après ce point de vue.

Le modèle d'informations de cœur de politique maintient un équilibre entre la complexité et le manque de puissance en utilisant les concepts logiques bien compris de forme normale disjonctive et de forme normale conjonctive pour combiner les conditions de politique simples en conditions plus complexes.

3. Vue d'ensemble du modèle d'information de cœur de politique

Le diagramme qui suit donne une vue d'ensemble des cinq classes centrales que comporte le modèle d'informations de cœur de politique, leurs associations les unes avec les autres, et leurs associations aux autres classes dans le modèle CIM global. Noter que la classe abstraite Policy et les deux classes d'extension VendorPolicyCondition et VendorPolicyAction ne sont pas montrées.

Note : "*" est l'abréviation de "0..n".

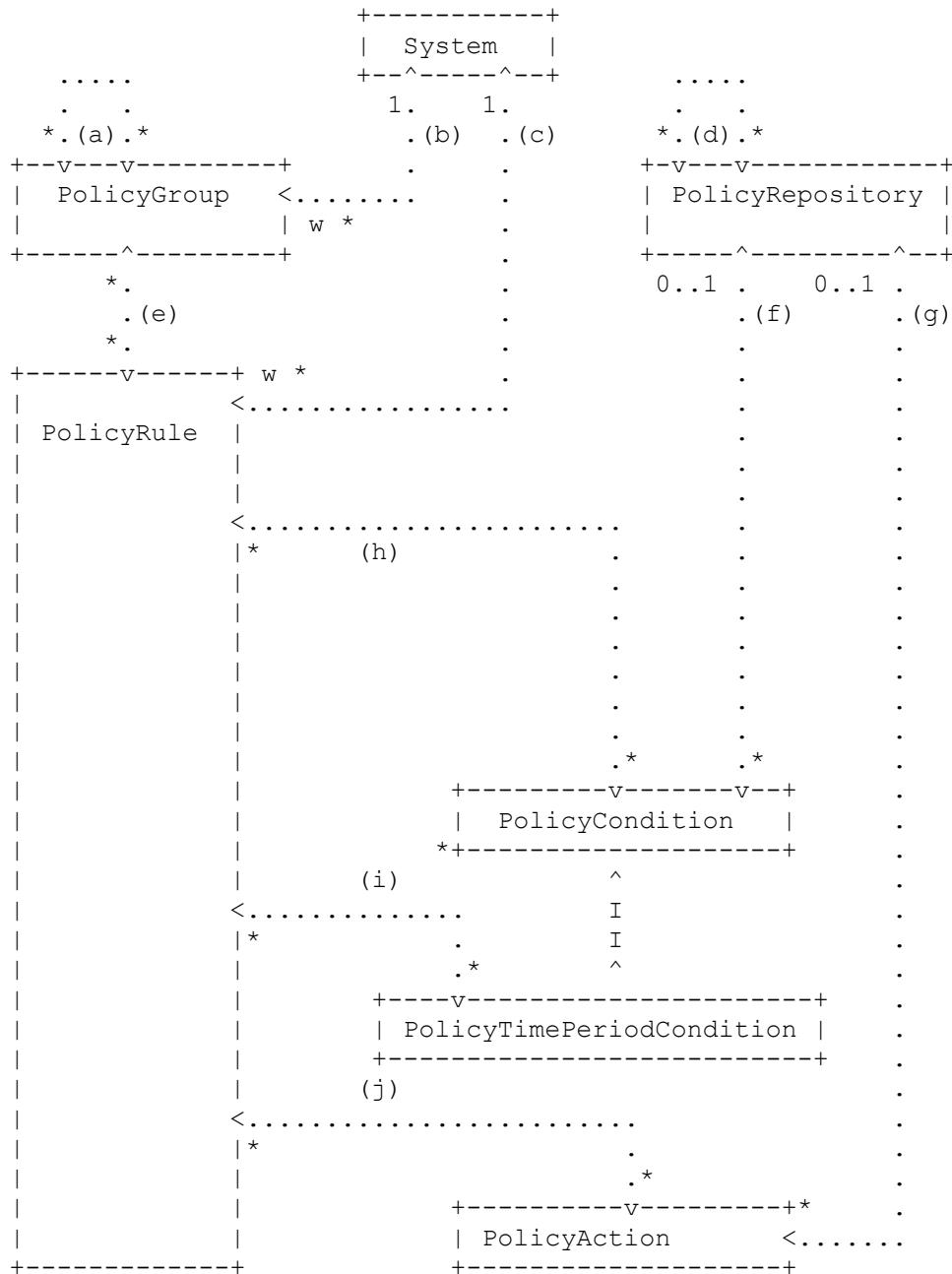


Figure 1 : Vue d'ensemble des classes de cœur de politique et de leurs relations

Dans cette figure les boîtes représentent les classes, et les flèches de points représentent les associations. Les associations suivantes apparaissent :

- (a) PolicyGroupInPolicyGroup
- (b) PolicyGroupInSystem
- (c) PolicyRuleInSystem
- (d) PolicyRepositoryInPolicyRepository
- (e) PolicyRuleInPolicyGroup
- (f) PolicyConditionInPolicyRepository
- (g) PolicyActionInPolicyRepository
- (h) PolicyConditionInPolicyRule
- (i) PolicyRuleValidityPeriod
- (j) PolicyActionInPolicyRule

Une association connecte toujours deux classes. Les "deux" classes peuvent cependant être la même classe, comme c'est le cas avec l'association PolicyGroupInPolicyGroup, qui représente l'inclusion récurrente de PolicyGroup dans d'autres PolicyGroup. L'association PolicyRepositoryInPolicyRepository est récurrente de la même façon.

Une association inclut des cardinaux pour chaque classe concernée. Ces cardinaux indiquent combien d'instances de chaque classe peuvent se rapporter à une instance de l'autre classe. Par exemple, l'association PolicyRuleInPolicyGroup a la gamme de cardinaux "*" (c'est-à-dire, "0..n") pour les deux classes PolicyGroup et PolicyRule. Ces gammes sont interprétées comme suit :

- o Une "*" après PolicyGroup indique qu'une PolicyRule peut n'être en relation avec aucun PolicyGroup, à un PolicyGroup, ou à plus d'un PolicyGroup via l'association PolicyRuleInPolicyGroup. En d'autres termes, une PolicyRule peut être contenue dans aucun PolicyGroup, dans un PolicyGroup, ou dans plus d'un PolicyGroup.
- o Une "*" après une PolicyRule indique qu'un PolicyGroup peut se rapporter à aucune PolicyRule, à une PolicyRule, ou à plus d'une PolicyRule via l'association PolicyRuleInPolicyGroup. En d'autres termes, un PolicyGroup peut contenir aucune PolicyRule, une PolicyRule, ou plus d'une PolicyRule.

Le "w" écrit après le PolicyGroupInSystem et PolicyRuleInSystem indique que ceci est ce que CIM appelle "agrégations avec références faibles", ou plus brièvement, "agrégations faibles". Une agrégation faible est simplement une indication d'une portée de dénomination. Donc, ces deux agrégations indiquent qu'une instance d'un PolicyGroup ou PolicyRule est désignée au sein de la portée d'un objet System. Une agrégation faible a implicitement la cardinalité 1..1 à l'extrémité opposée au 'w'.

Les associations montrées à la Figure 1 sont expliquées plus en détail à la Section 7.

4. Hiérarchies héritées pour le modèle d'information de cœur de politique

Le diagramme qui suit illustre la hiérarchie d'héritage pour les classes de cœur de politique :

```

ManagedElement (abstrait)
|
+---Policy (abstrait)
| |
| | +---PolicyGroup
| | |
| | +---PolicyRule
| | |
| | +---PolicyCondition (abstrait)
| | |
| | | +---PolicyTimePeriodCondition
| | | |
| | | +---VendorPolicyCondition
| | | |
| | +---PolicyAction (abstrait)
| | |
| | +---VendorPolicyAction
| |
+---ManagedSystemElement (abstrait)
|
+---LogicalElement (abstrait)
|
+---System (abstrait) |
|
+---AdminDomain (abstrait)
|
+---PolicyRepository

```

Figure 2 : Hiérarchie d'héritage pour les classes de cœur de politique

ManagedElement, ManagedSystemElement, LogicalElement, System, et AdminDomain sont définis dans le schéma CIM [1]. Ces classes ne sont pas discutées en détail dans le présent document.

Dans CIM, les associations sont aussi modélisées comme des classes. Pour le modèle d'informations de cœur de politique, la hiérarchie d'héritage pour les associations est la suivante :

```

[pas de racine]
|
+---PolicyComponent (abstrait)
|   |
|   +---PolicyGroupInPolicyGroup
|   |
|   +---PolicyRuleInPolicyGroup
|   |
|   +---PolicyConditionInPolicyRule
|   |
|   +---PolicyRuleValidityPeriod
|   |
|   +---PolicyActionInPolicyRule
|
+---Dependency (abstrait)
|   |
|   +---PolicyInSystem (abstrait)
|   |
|   |   +---PolicyGroupInSystem
|   |   |
|   |   +---PolicyRuleInSystem
|   |   |
|   |   +---PolicyConditionInPolicyRepository
|   |   |
|   |   +---PolicyActionInPolicyRepository
|   |
|
+---Component (abstrait)
|   |
|   +---SystemComponent
|   |
|   +---PolicyRepositoryInPolicyRepository

```

Figure 3 : Hiérarchie d'héritage pour les associations de cœur de politique

Les associations Dependency, Component, et SystemComponent sont définies dans le schéma CIM [1], et ne sont pas discutées dans le présent document.

4.1 Implications de l'héritage CIM

À partir du schéma CIM, les propriétés et associations sont toutes deux héritées des classes Policy. Par exemple, la classe ManagedElement est référencée dans les associations Dependency, Statistics et MemberOfCollection. Et l'association Dependency est à son tour référencée dans l'association DependencyContext. À ce niveau très abstrait et élevé dans la hiérarchie d'héritage, le nombre de ces associations est très petit et leur sémantique est assez générale.

Beaucoup de ces associations héritées portent une sémantique supplémentaire qui n'est pas nécessaire pour la compréhension du modèle d'informations de cœur de politique. En fait, elle est définie comme FACULTATIVE dans le schéma CIM – car sa cardinalité est "0..n" sur toutes les références. Le document PCIM discute spécifiquement de ce qui est nécessaire pour la prise en charge et l'instanciation. Par exemple, à travers les sous-classes de l'association Dependency est décrite la sémantique exacte de Dependency dans PCIM.

Ainsi, on peut se demander que faire de ces autres associations héritées. La réponse est "les ignorer sauf si on a besoin d'elles". On pourrait en avoir besoin pour décrire des informations et une sémantique supplémentaires pour des données de politique. Par exemple, elles peuvent être nécessaires pour capturer des données statistiques pour une PolicyRule (pour la règle dans un répertoire ou pour l'exécution d'un système de politique). Des exemples de données statistiques pour une règle sont le nombre de fois qu'elle a été téléchargée, le nombre de fois que ses conditions ont été évaluées, et le nombre de fois que ses actions ont été exécutées. (Ces types de données seraient décrits dans une sous classe de CIM_StatisticalInformation.) Dans ce cas, l'association Statistics héritée de ManagedElement pour PolicyRule peut être utilisée pour décrire le lien entre une instance d'une PolicyRule et l'ensemble de statistiques pour elle.

5. Détails du modèle

Les paragraphes qui suivent exposent plusieurs questions spécifiques en rapport avec le modèle d'informations de cœur de politique.

5.1 Conditions et actions réutilisables ou spécifiques de règles

Les conditions et les actions de politique peuvent être partagées en deux groupes : les unes sont associées à une seule règle de politique, et les autres sont réutilisables, en ce sens qu'elles peuvent être associées à plus d'une règle de politique. Les conditions et actions du premier groupe sont appelées conditions et actions "spécifiques d'une règle" ; celles du second groupe sont caractérisées comme "réutilisables".

Il est important de comprendre que la différence entre une condition ou action qui est spécifique d'une règle et une qui est réutilisable se fonde sur l'intention de l'administrateur de politique pour la condition ou action, plutôt que sur les associations actuelles auxquelles participe la condition ou action. Donc, une condition ou action réutilisable (c'est-à-dire, une qu'un administrateur de politique a créé pour être réutilisable) peut à un moment donné être associée à exactement une règle de politique, sans devenir par là spécifique d'une règle.

Il n'y a pas de différence inhérente entre la condition ou action spécifique d'une règle et celle qui est réutilisable. Il y a cependant des différences dans la façon dont elles sont traitées dans un répertoire de politique. Par exemple, il est naturel de donner les permissions d'accès pour une condition ou action spécifique d'une règle identique à celles pour la règle elle-même. Il est aussi naturel qu'une condition ou action spécifique d'une règle soit retirée du répertoire de politique au même moment que l'est la règle. D'un autre côté, avec des conditions et actions réutilisables, les permissions d'accès et les critères d'existence doivent être exprimables sans référence à une règle de politique.

Le paragraphe précédent ne contient pas une liste exhaustive des façons dont les conditions réutilisables et spécifiques d'une règle devraient être traitées différemment. Son objet est simplement de justifier de faire une distinction sémantique entre spécifique d'une règle et réutilisable, et ensuite de refléter cette distinction dans le modèle de politique lui-même.

Un problème est mis en lumière par les conditions et actions de politique réutilisables et spécifiques d'une règle : le manque d'une capacité programmatique pour exprimer des contraintes complexes impliquant plusieurs associations. En prenant PolicyCondition comme exemple, il y a deux agrégations à regarder. PolicyConditionInPolicyRule a la cardinalité * aux deux extrémités, et PolicyConditionInPolicyRepository a la cardinalité * à l'extrémité PolicyCondition, et [0..1] à l'extrémité PolicyRepository.

Globalement, ces cardinalités sont correctes. Cependant, cette histoire n'est pas terminée, qui ne devient claire que si on examine les cardinalités séparément pour les deux cas d'une PolicyCondition spécifique d'une règle et d'une réutilisable.

Pour une PolicyCondition spécifique de règle, la cardinalité de PolicyConditionInPolicyRule à l'extrémité PolicyRule est [1..1], plutôt que [0..n] (on rappelle que * est une abréviation pour [0..n]) car la condition est unique pour une règle de politique. Et la cardinalité de PolicyConditionInPolicyRepository à l'extrémité PolicyRepository est [0..0], car la condition n'est pas dans le répertoire "réutilisable". Ceci est correct, car ce sont deux sous ensembles des cardinalités spécifiées.

Cependant, pour une PolicyCondition réutilisable, la cardinalité de PolicyConditionInPolicyRepository à l'extrémité PolicyRepository est [1..1], car la condition doit être dans le répertoire. Et la cardinalité de PolicyConditionInPolicyRule à l'extrémité PolicyRule est [0..n]. Ce dernier point est important : une PolicyCondition réutilisable peut être associée à 0, 1, ou plus d'une PolicyRule, via exactement la même association PolicyConditionInPolicyRule qui lie une condition spécifique d'une règle à sa PolicyRule.

Actuellement, la seule façon de documenter les contraintes de ce type est textuellement. Des méthodes plus formelles pour documenter les contraintes complexes sont nécessaires.

5.2 Rôles

5.2.1 Rôles et combinaisons de rôles

Le concept de rôle est central pour la conception du cadre entier de politique. L'idée derrière les rôles est simple. Plutôt que de configurer, et ensuite d'avoir une mise à jour de la configuration, des centaines ou des milliers (ou plus) de ressources dans un réseau, un administrateur de politique alloue chaque ressource à un ou plusieurs rôles, et ensuite spécifie les politiques pour chacun de ces rôles. Le cadre de politique est alors chargé de configurer chacune des ressources associées à

un rôle d'une façon telle qu'elle se comporte conformément aux politiques spécifiées pour ce rôle. Lorsque le comportement du réseau doit être changé, l'administrateur de politique peut effectuer une seule mise à jour de la politique pour un rôle, et le cadre de politique va assurer que les mises à jour de configuration nécessaires sont effectuées sur toutes les ressources qui tiennent ce rôle.

Une définition plus formelle d'un rôle est la suivante :

Un rôle est un type d'attribut qui est utilisé pour choisir une ou plusieurs politiques pour un ensemble d'entités et/ou composants parmi un ensemble beaucoup plus grand de politiques disponibles.

Des rôles peuvent être combinés ensemble. Voici une définition formelle d'une "combinaison de rôles" :

Une combinaison de rôles est un ensemble d'attributs qui sont utilisés pour choisir une ou plusieurs politiques pour un ensemble d'entités et/ou composants dans un ensemble beaucoup plus grand de politiques disponibles. Comme l'illustrent les exemples ci-dessous, le processus de choix pour une combinaison de rôles choisit les politiques associées à la combinaison elle-même, les politiques associées à chacune de ses sous combinaisons, et les politiques associées à chacun des rôles individuels de la combinaison de rôles.

Il est important de noter qu'un rôle est plus qu'un attribut. Un rôle définit une fonction particulière d'une entité ou composant qui peut être utilisé pour identifier un comportement particulier associé à cette entité ou composant. Cette différence est critique, et elle est très facilement comprise en pensant à un rôle comme à un sélecteur. Lorsque il est utilisé de cette manière, un rôle (ou combinaison de rôle) choisit un ensemble de politiques différent de celui qu'aurait choisi un rôle (ou combinaison de rôles) différent.

Les rôles et combinaisons de rôles sont particulièrement utiles pour choisir quelles politiques sont applicables à un ensemble particulier d'entités ou composants lorsque le répertoire de politiques peut mémoriser des centaines ou des milliers de politiques. Cette utilisation souligne la capacité du rôle (ou combinaison de rôles) à sélectionner le petit sous ensemble de politiques qui sont applicables dans un énorme ensemble de politiques disponibles.

Un exemple va illustrer comment fonctionnent réellement les combinaisons de rôles. Supposons qu'une installation ait trois rôles définis pour ses interfaces : "Ethernet", "Campus", et "WAN". Dans le répertoire de politiques, certaines règles de politique pourraient être associées au rôle "Ethernet" ; ces règles s'appliqueraient à toutes les interfaces Ethernet, sans considérer si elles sont sur le côté campus ou sur le côté WAN. D'autres règles pourraient être associées à la combinaison de rôles "Campus"+"Ethernet" ; ces règles s'appliqueraient au côté campus des interfaces Ethernet, mais pas à celles sur le côté WAN. Finalement, un troisième jeu de règles pourrait être associé à la combinaison de rôles "Ethernet"+"WAN" ; ces règles s'appliqueraient aux interfaces Ethernet côté WAN, mais pas à celles du côté campus. (Les rôles dans une combinaison de rôles apparaissent en ordre alphabétique dans ces exemples, parce que c'est comme cela qu'elles apparaissent dans le modèle d'informations.)

Si on a une interface spécifique A qui est associée à la combinaison de rôles "Ethernet"+"WAN", on voit qu'elle devrait se voir appliquer trois catégories de règles de politique : celles pour le rôle "Ethernet", celles pour le rôle "WAN", et celles pour la combinaison de rôles "Ethernet"+"WAN". Allons un pas plus loin : si l'interface B est associée à la combinaison de rôles "département"+"Ethernet"+"WAN", alors B devrait avoir sept catégories de règles de politique qui s'appliquent à elle - celles associées aux combinaisons de rôles suivantes :

- o "département"
- o "Ethernet"
- o "WAN"
- o "département"+"Ethernet"
- o "département"+"WAN"
- o "Ethernet"+"WAN"
- o "département"+"Ethernet"+"WAN".

Afin d'obtenir toutes les bonnes règles de politique pour une ressource comme l'interface B, un PDP doit étendre la seule combinaison de rôle qu'il reçoit pour B en cette liste de sept combinaisons de rôle, et ensuite restituer à partir du répertoire de politiques les sept ensembles correspondants de règles de politique. Bien sûr, cet exemple est plus compliqué que la normale : un cas normal va impliquer d'étendre une combinaison de deux rôles en trois valeurs identifiant ces trois ensembles de règles de politique.

Les combinaisons de rôle aident aussi à simplifier un peu le problème de l'identification des conflits entre règles de politique. Avec les combinaisons de rôles, il est possible à un administrateur de politique de spécifier un ensemble de règles de politique pour les interfaces Ethernet côté campus, et un second ensemble de règles de politique pour les interfaces Ethernet côté WAN, sans avoir à se soucier des conflits entre les deux jeux de règles. L'administrateur de politique "désactive" simplement la détection de conflit pour ces deux ensembles de règles, en disant au système de gestion de politique que les rôles "Campus" et "WAN" sont incompatibles l'un avec l'autre. Cela indique que la combinaison de rôles ne va jamais se produire, et il n'y aura donc jamais de conflit. Dans certains cas, la technologie elle-même peut identifier

des rôles incompatibles, par exemple, "Ethernet" et "FrameRelay". Mais avec des termes moins précis comme "Campus" et "WAN", l'administrateur de politique doit dire si ils identifient des rôles incompatibles.

Lorsque l'administrateur de politique fait cela, il y a trois effets :

1. Si une interface s'est vu allouer une combinaison de rôles impliquant à la fois "Campus" et "WAN", le système de gestion de politique peut le signaler comme erreur.
2. Si une règle de politique est associée à une combinaison de rôles impliquant à la fois "Campus" et "WAN", le système de gestion de politique peut le signaler comme erreur.
3. Si le système de gestion de politique voit deux règles de politique où l'une est liée au rôle "Campus" (ou à une combinaison de rôles qui inclut le rôle "Campus") et où l'autre est liée au rôle "WAN" (ou à une combinaison de rôles qui inclut le rôle "WAN") alors le système n'a pas besoin de chercher des conflits entre les deux règles de politique : parce que les rôles sont incompatibles, les deux règles ne peuvent pas entrer en conflit.

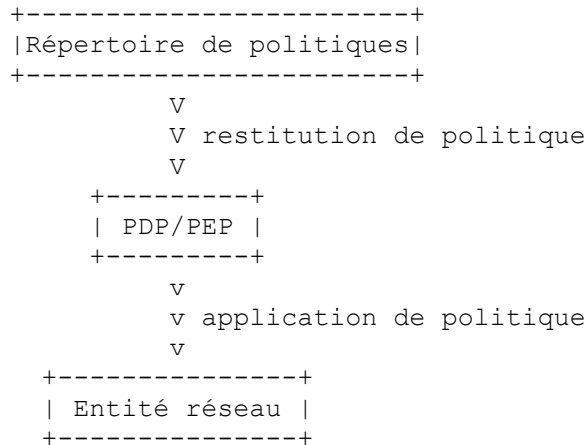


Figure 4 : Restitution et application d'une politique

La Figure 4, qui n'est donnée qu'à titre d'exemple de la façon dont le cadre de politique pourrait être mis en œuvre par une collection de composants de réseau, illustre comment les rôles fonctionnent au sein du cadre de politique. Comme la distinction entre eux n'a pas d'importance pour notre exposé, le PDP et le PEP sont combinés en une boîte. Les points qu'on illustre ici s'appliquent cependant également bien à un environnement où le PDP et le PEP sont mis en œuvre séparément.

Un rôle représente une caractéristique fonctionnelle ou une capacité d'une ressource à laquelle s'appliquent des politiques. Des exemples de rôles incluent une interface de cœur de réseau, une interface de relais de trame, un routeur à capacité BGP, un serveur de la Toile, un pare-feu, etc. Les multiples rôles alloués à une seule ressource sont combinés pour former la combinaison de rôles de cette ressource. Les combinaisons de rôle sont représentées dans le PCIM par des valeurs de la propriété PolicyRoles dans la classe PolicyRule. Un PDP utilise les rôles de politique comme suit pour identifier les politiques qu'il a besoin de connaître :

1. Le PDP apprend d'une certaine façon la liste des rôles que tiennent ses PEP. Ces informations peuvent être configurées au PDP, les PEP peuvent les fournir au PDP, ou le PDP peut aller les chercher dans un répertoire.
2. En utilisant des moyens spécifiques du répertoire, le PDP détermine où chercher les règles de politique qu'il peut lui appliquer.
3. En utilisant les rôles et combinaisons de rôles qu'il reçoit de ses PEP comme indiqué dans les exemples ci-dessus, le PDP est capable de localiser et restituer les règles de politique qui sont pertinentes pour lui.

5.2.2 Propriétés de PolicyRoles

Comme on l'a indiqué plus haut, PolicyRoles est une propriété associée à une règle de politique. C'est une matrice qui contient des "combinaisons de rôles" pour la règle de politique, et qui les corrèle aux rôles définis pour une ressource du réseau. En utilisant la propriété PolicyRoles, il est possible de marquer une règle de politique comme s'appliquant, par exemple, à une interface de relais de trame ou à une interface de cœur de réseau ATM. La propriété PolicyRoles prend des chaînes de la forme :

```
<RoleName>[&&<RoleName>]*
```

Chaque valeur de cette propriété représente une combinaison de rôles, y compris le cas particulier d'une "combinaison" ne contenant qu'un rôle. Comme le format l'indique, les noms de rôle dans une combinaison de rôles sont ajoutés les uns aux autres pour former un seul sélecteur. Les valeurs multiples de la propriété PolicyRoles sont logiquement OUixées, pour

rendre possible qu'une règle de politique ait plusieurs sélecteurs.

Les noms de rôle individuels dans une combinaison de rôles doivent apparaître en ordre alphabétique (selon la séquence de collation des caractères UCS-2) pour faire fonctionner correctement les correspondances de chaîne. Les noms de rôle utilisés dans un environnement sont spécifiés par l'administrateur de politique.

5.3 Heure locale et UTC dans PolicyTimePeriodConditions

Une instance de PolicyTimePeriodCondition a jusqu'à cinq propriétés qui représentent les temps : TimePeriod, MonthOfYearMask, DayOfMonthMask, DayOfWeekMask, et TimeOfDayMask. Toutes les propriétés qui se rapportent au temps dans une instance de PolicyTimePeriodCondition représentent un des deux types d'heure : heure locale à l'endroit où une règle de politique est appliquée, ou temps UTC. La propriété LocalOrUtcTime indique quelle représentation temporelle s'applique à une instance de PolicyTimePeriodCondition.

Comme le PCIM ne fournit que l'heure locale et le temps UTC, un outil de gestion de politique qui fournit d'autres représentations de l'heure (par exemple, une heure fixée à un endroit particulier) aura besoin de se transposer de ces autres représentations soit en heure locale soit en temps UTC. Un exemple va illustrer la nature de cette transposition.

Supposons qu'une règle de politique soit liée aux heures de fonctionnement d'un service d'assistance : de 8 h à 20 h du lundi au vendredi, heure de l'Est américain. Pour exprimer ces temps dans PolicyTimePeriodCondition, un outil de gestion doit les convertir en UTC. (Ce ne sont pas des heures locales, parce qu'elles se réfèrent à un seul intervalle de temps au niveau mondial et non pas aux intervalles liés aux horloges locales aux endroits où la PolicyRule est appliquée.) Comme le montre la [RFC2445], la transposition de l'heure de l'Est américain en temps UTC n'est pas simplement une affaire d'application d'un décalage : le décalage entre l'heure de l'Est américain et le temps UTC change entre - 0500 et - 0400 selon que l'heure d'été ou d'hiver s'applique aux USA.

Supposons que le but de l'administrateur de politique soit d'avoir une règle de politique valide de 0800 à 1200 en heure Est américaine tous les lundis, dans la période de temps allant du début 2000 à la fin 2001. L'outil de gestion de politique pourrait être configuré avec la définition de ce que signifie l'heure de l'Est américain, ou il pourrait être configuré avec la connaissance de l'endroit où aller pour obtenir cette information. La [RFC2445] contient un exposé plus détaillé des définitions de zone horaire et de l'endroit où elles se trouvent.

Armé de ses connaissances sur l'heure de l'Est américain, l'outil de gestion de politique va créer le nombre d'instances de PolicyTimePeriodCondition dont il a besoin pour représenter les intervalles désirés. Noter qu'alors qu'il y a un nombre accru d'instances de PolicyTimePeriodCondition, il n'y a qu'une seule PolicyRule, qui est liée à toutes les instances de PolicyTimePeriodCondition via l'agrégation PolicyRuleValidityPeriod. Voici les deux premières de ces instances :

1. TimePeriod: 20000101T050000/20000402T070000
 DayOfWeekMask: { lundi }
 TimeOfDayMask: T130000/T170000
 LocalOrUtcTime: UTC
2. TimePeriod: 20000402T070000/20001029T070000
 DayOfWeekMask: { lundi }
 TimeOfDayMask: T120000/T160000
 LocalOrUtcTime: UTC

Il y aurait trois instances similaires supplémentaires, pour hiver 2000-2001, été 2001, et hiver 2001 jusqu'au 31 décembre.

Si l'exemple avait été choisi différemment, il aurait pu y avoir encore plus d'instances de PolicyTimePeriodCondition. Si, par exemple, l'intervalle de temps avait été de 0800 - 2200 en heure de l'Est américain le lundi, l'instance 1 ci-dessus se serait partagée en deux instances : une avec un intervalle de temps UTC de T130000/T240000 les lundis, et un autre avec un intervalle de temps UTC de T000000/T030000 les mardis. De sorte que le résultat final aurait été de dix instances de PolicyTimePeriodCondition, et pas de cinq.

En restreignant PolicyTimePeriodCondition à l'heure locale et au temps UTC, le PCIM déplace la tâche difficile et coûteuse de transposition des représentations d'heure "humaine" en représentations lisibles par la machine dans l'outil de gestion de politique. Une autre approche aurait été de placer dans PolicyTimePeriodCondition un moyen de représenter une zone horaire désignée, comme l'heure de l'Est américain. Cela aurait cependant passé la responsabilité de la difficile transposition aux PDP et PEP. Il vaut mieux avoir une transposition comme celle décrite ci-dessus qui est faite une fois dans un outil de gestion de politique, que d'en avoir une qui est faite encore et toujours dans chacun des PDP (et éventuellement des PEP) qui a besoin d'appliquer une PolicyRule.

5.4 Types de données CIM

Comme PCIM étend le schéma CIM, une correspondance est nécessaire entre les types de données utilisés dans CIM et dans PCIM. Les types de données CIM suivants sont utilisés dans les définitions de classes qui suivent aux Sections 6 et 7 :

- o uint8 entier non signé de 8 bits
- o uint16 entier non signé de 16 bits
- o boolean booléen
- o string chaîne UCS-2.

Les chaînes dans CIM sont mémorisées par des caractères UCS-2, où chaque caractère est codé sur deux octets. Donc les valeurs de chaînes peuvent devoir être converties lors d'un passage d'un environnement CIM à un environnement qui utilise un codage de chaîne différent. Par exemple, dans un répertoire accessible par LDAP, les attributs de type DirectoryString sont mémorisés en format UTF-8. La [RFC2279] explique comment faire la conversion entre ces deux formats.

Lorsque elle est appliquée à une chaîne CIM, une valeur MaxLen se réfère au nombre maximum de caractères de la chaîne, plutôt qu'au nombre maximum d'octets.

En plus des types de données CIM énumérés ci-dessus, les associations de classes de la Section 7 utilisent le type suivant :

- o <classname> ref référence fortement typée.

Il y a une omission évidente dans cette liste des types de données CIM : chaîne d'octets. C'est parce que CIM traite les chaînes d'octets comme un type de données dérivé. Il y a deux formes de chaînes d'octets dans CIM - un dispositif uint8 ordonné pour les chaînes d'une seule valeur, et un dispositif de chaîne pour les propriétés multi valeurs. Toutes deux sont décrites en ajoutant un qualificatif "OctetString" (méta-données) à la propriété. Ce qualificatif fonctionne exactement comme une convention textuelle SMIV2 (SNMP) précisant la syntaxe et la sémantique du type de données CIM existant.

Les quatre premiers éléments numériques des deux représentations de "OctetString" sont un champ de longueur. (La raison pour laquelle l'adjectif "numérique" est ajouté à la phrase précédente est que la propriété chaîne inclut aussi '0' et 'x', comme premiers caractères.) Dans les deux cas, ces quatre éléments numériques (octets) sont inclus dans le calcul de la longueur. Par exemple, une propriété de chaîne d'octet d'une seule valeur ayant la valeur X'7C' serait représentée par le dispositif uint8 X'00 00 00 05 7C'.

Les chaînes qui représentent les valeurs individuelles d'une propriété multi valeurs qualifiée avec "OctetString" sont construites de façon similaire :

1. Prendre une valeur à coder comme une chaîne d'octets (on utilise X'7C' comme précédemment) et on y ajoute devant une longueur de quatre octets. Le résultat est le même, X'00 00 00 05 7C'.
2. Convertir cela en une chaîne de caractères en introduisant devant '0' et 'x', et retirer toutes les espaces. Nous avons donc la chaîne de douze caractères "0x000000057C". Cette chaîne est la valeur d'un des éléments du dispositif dans la matrice de chaîne CIM. Comme CIM utilise le jeu de caractères UCS-2, il va falloir 24 octets pour coder cette chaîne de 12 caractères.

Les transpositions de PCIM en un modèle de données particulier ne sont pas obligées de suivre cette technique CIM de représentation des chaînes d'octets multi valeurs comme des chaînes de caractères de longueur préfixées. Dans une transposition LDAP, par exemple, il serait beaucoup plus naturel de simplement utiliser la syntaxe de CHAINE D'OCTETS, et d'omettre les octets de longueur en préfixe.

5.5 Comparaison entre spécifications de classe CIM et LDAP

Il y a un certain nombre de différences entre les spécifications de classe de CIM et de LDAP. Celles qui sont pertinentes pour les spécifications de classe abrégées du présent document sont énumérées ci-dessous. Ces éléments sont inclus ici pour aider la compréhension de la communauté de l'IETF, qui est déjà familiarisée avec LDAP, à la modélisation CIM, et par extension, à la modélisation d'informations en général.

- o Au lieu des trois types de classes de LDAP (abstrait, auxiliaire, structurel) CIM en a seulement deux : abstrait et instanciable. Le type d'une classe CIM est indiqué par le qualificatif booléen ABSTRAIT.
- o CIM utilise le terme de "propriété" pour ce que LDAP appelle un "attribut".

- o CIM utilise la notation "[]" pour indiquer qu'une propriété est multi valeurs CIM définit trois types de dispositifs : sacs (les contenus ne sont pas ordonnés, les dupliqués sont permis) sacs ordonnés (les contenus sont ordonnés mais les dupliqués sont permis) et dispositif indexé (les contenus sont ordonnés et aucun dupliqué n'est admis).
- o Les classes et propriétés CIM sont identifiées par le nom, et non par un OID.
- o Les classes CIM utilisent un schéma de dénomination différent pour les mises en œuvre natives, de celui de LDAP. Le schéma de dénomination de CIM est documenté à l'Appendice A car il n'est pas critique de comprendre le modèle d'informations, et il ne s'applique que lors d'une communication avec une mise en œuvre CIM native.
- o Dans LDAP, les définitions d'attribut sont globales, et le même attribut peut apparaître dans plusieurs classes. Dans CIM, une propriété est définie au sein de la portée d'une seule définition de classe. La propriété peut être héritée par des sous-classes de la classe dans laquelle elle est définie, mais autrement, elle ne peut pas apparaître dans d'autres classes. Un effet collatéral de cette différence est que les noms de propriété CIM tendent à être beaucoup plus courts que les noms d'attribut LDAP, car leur portée est implicitement donnée par le nom de la classe dans laquelle ils sont définis.

Il y a aussi une convention de notation que suit le présent document pour améliorer la lisibilité : dans CIM, tous les noms de classe et de propriété sont préfixés par les caractères "CIM_". Ces préfixes ont été omis dans le présent document, à une exception près concernant la désignation, documentée dans l'Appendice A.

Pour la définition complète du langage de spécification CIM, voir la référence [2].

6. Définitions de classe

Les paragraphes qui suivent contiennent les définitions des classes PCIM.

6.1 Classe abstraite "Policy"

La classe abstraite Policy collecte plusieurs propriétés qui peuvent être incluses dans des instances de toute classe de cœur de politique (ou leurs sous-classes). Pour simplifier, les deux propriétés qu'hérite Policy de ManagedElement dans le schéma CIM sont aussi présentées ici.

La définition de classe est la suivante :

```
NOM : Policy
DESCRIPTION : classe abstraite avec quatre propriétés pour décrire une instance en rapport avec la politique.
DÉDUIT DE : ManagedElement
ABSTRAIT : VRAI
PROPRIÉTÉS : CommonName (CN)
              PolicyKeywords[ ]
              // Caption (héritée)
              // Description (héritée)
```

6.1.1 Propriété "CommonName (CN)"

La propriété CN, ou CommonName, correspond à l'attribut X.500 commonName (cn). Dans X.500, cette propriété spécifie un ou plusieurs noms faciles à mémoriser par l'utilisateur (normalement, un seul nom) par lequel l'objet est couramment désigné, des noms qui se conforment aux conventions de désignation du pays ou culture auquel l'objet est associé. Cependant, dans le modèle CIM, la propriété CommonName a une seule valeur.

```
NOM : CN
DESCRIPTION : nom facile à mémoriser par l'utilisateur d'un objet en rapport avec la politique.
SYNTAXE : chaîne
```

6.1.2 Propriété multi valeurs "PolicyKeywords"

Cette propriété fournit un jeu d'un ou plusieurs mots-clés qu'un administrateur de politique peut utiliser pour aider à caractériser ou catégoriser un objet de politique. Les mots-clés sont de l'un des deux types suivants :

- o Mots-clés définis dans ce document, ou dans des documents qui définissent des sous classes des classes définies dans ce document. Ces mots-clés fournissent un moyen indépendant du fabricant et de l'installation pour caractériser des

objets de politique.

- o Mots-clés dépendants de l'installation pour caractériser les objets de politique. Des exemples incluent "ingénierie", "facturation", et "Révision en décembre 2000".

Le présent document définit les mots-clés suivants : "INCONNU", "CONFIGURATION", "USAGE", "SÉCURITÉ", "SERVICE", "MOTIVANT", "INSTALLATION", et "ÉVÉNEMENT". Ces concepts ont été définis à la Section 2.

Un mot-clé supplémentaire est défini : "POLITIQUE". Le rôle de ce mot-clé est d'identifier les instances en rapport avec la politique qui ne seraient autrement pas identifiables comme étant en rapport avec la politique. Il peut être nécessaire dans certaines mises en œuvre de répertoire.

Les documents qui définissent des sous classes de classes du modèle d'informations de cœur de politique DEVRAIENT définir des mots-clés supplémentaires pour caractériser les instances de ces sous classes. Par convention, les mots-clés définis en conjonction avec les définitions de classe sont en majuscules. Les mots-clés définis par l'installation peuvent être dans l'une ou l'autre casse.

La définition de la propriété est la suivante :

NOM : PolicyKeywords
 DESCRIPTION : ensemble de mots-clés pour caractériser/catégoriser des objets de politique.
 SYNTAXE : chaîne

6.1.3 Propriété "Caption" (héritée de ManagedElement)

Cette propriété fournit une description sur une ligne d'un objet en rapport avec la politique.

NOM : Caption
 DESCRIPTION : description sur une ligne de cet objet en rapport avec la politique.
 SYNTAXE : chaîne

6.1.4 Propriété "Description" (héritée de ManagedElement)

Cette propriété donne une plus longue description que celle fournie par la propriété caption.

NOM : Description
 DESCRIPTION : longue description de cet objet en rapport avec la politique.
 SYNTAXE : chaîne

6.2 Classe "PolicyGroup"

Cette classe est un conteneur d'agrégation généralisé. Il permet d'agréger des PolicyRule ou des PolicyGroup en un seul conteneur. Les boucles, y compris le cas anormal d'un PolicyGroup qui se contient lui-même, ne sont pas permises lorsque les PolicyGroup contiennent d'autres PolicyGroup.

Les PolicyGroup et leurs capacités d'hébergement sont montrés à la Figure 5. Noter qu'un PolicyGroup peut héberger d'autres PolicyGroup, et il n'y a pas de restriction sur la profondeur de l'hébergement de PolicyGroup parents.

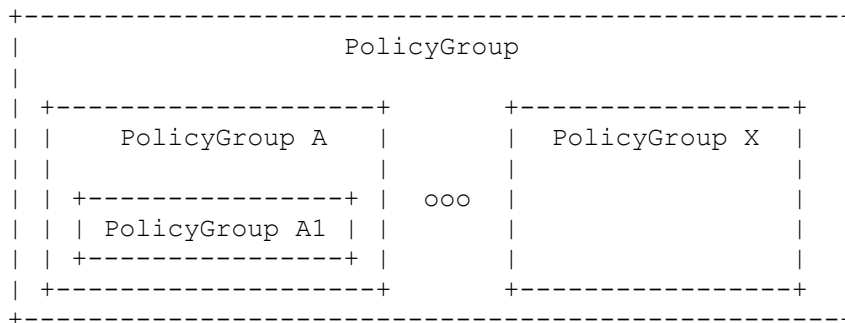


Figure 5 : Vue d'ensemble de la classe PolicyGroup

Pour un simple exemple, voyez le plus haut niveau de PolicyGroup de la Figure 5 ci-dessus comme la politique de

connexion des employés américains d'une société. Ce PolicyGroup peut être appelé USEmployeeLogonPolicy, et peut agréger plusieurs PolicyGroup qui donnent des règles spécialisées selon la localisation. Donc, le PolicyGroup A de la Figure 5 peut définir les règles de connexion pour les employés de la côte Ouest, tandis qu'un autre PolicyGroup pourrait définir les règles de connexion pour le Centre Ouest (par exemple, PolicyGroup X), et ainsi de suite.

Noter aussi que la profondeur de chaque PolicyGroup n'a pas besoin d'être la même. Donc, le PolicyGroup WestCoast peut avoir plusieurs couches supplémentaires de PolicyGroup définies pour plusieurs raisons (locaux différents, nombre de sous réseaux, etc.). Les PolicyRule sont donc contenues à n niveaux à partir du USEmployeeLogonPolicyGroup. Comparer ceci au PolicyGroup Midwest (PolicyGroup X) qui pourrait contenir directement les PolicyRule.

La définition de classe pour PolicyGroup est la suivante :

NOM : PolicyGroup
 DESCRIPTION : conteneur pour un ensemble de PolicyRule en rapport ou ensemble de PolicyGroup en rapports.
 DÉDUIT DE : Policy
 ABSTRAIT : FAUX
 PROPRIÉTÉS : AUCUNE

Aucune propriété n'est définie pour cette classe car elle hérite toutes ses propriétés de Policy. La classe existe pour agréger les PolicyRule ou d'autres PolicyGroup. Elle est directement instanciable. Dans une mise en œuvre, diverses propriétés clé/identification DOIVENT être définies. Les clés pour une mise en œuvre native de CIM sont définies à l'Appendice A, paragraphe 13.1.1. Les clés pour une mise en œuvre LDAP seront définies dans la transposition LDAP de ce modèle d'informations [11].

6.3 Classe "PolicyRule"

Cette classe représente la sémantique "Si condition alors action" associée à une politique. Une condition PolicyRule, au sens le plus général, est représentée soit comme un ensemble OUIxé de conditions ETixées (forme disjonctive normale, ou DNF) soit un ensemble ETixé de conditions OUIxées (forme conjonctive normale, ou CNF). Des conditions individuelles peuvent être soit niées (NON C) soit affirmées (C). Les actions spécifiées par une PolicyRule sont à effectuer si et seulement si la condition PolicyRule (selon qu'elle est représentée en DNF ou en CNF) s'évalue à VRAI.

Les conditions et actions associées à une règle de politique sont modélisées, respectivement, avec des sous classes des classes PolicyCondition et PolicyAction. Ces objets de condition et action sont liés aux instances de PolicyRule par les agrégations PolicyConditionInPolicyRule et PolicyActionInPolicyRule.

Comme illustré ci-dessus à la Section 3, une règle de politique peut aussi être associée à un ou plusieurs horaires de politique, indiquant le programme selon lequel la règle de politique est active et inactive. Dans ce cas, c'est l'agrégation PolicyRuleValidityPeriod qui fait la liaison.

Une règle de politique est illustrée conceptuellement ci-dessous par la Figure 6.

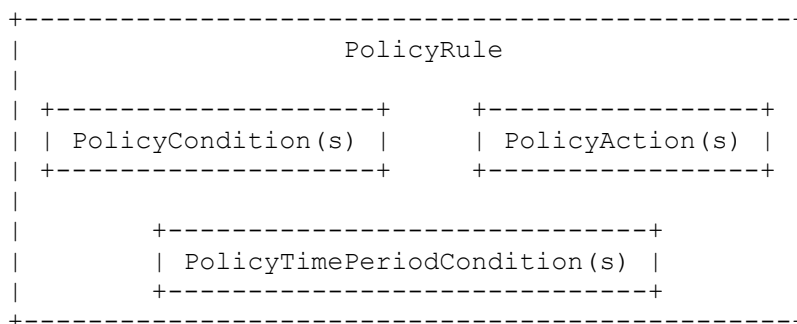


Figure 6 : Vue d'ensemble de la classe PolicyRule

La classe PolicyRule utilise la propriété ConditionListType, pour indiquer si les conditions pour la règle sont en DNF ou en CNF. L'agrégation PolicyConditionInPolicyRule contient deux propriétés supplémentaires pour compléter la représentation de l'expression conditionnelle de la règle. La première de ces propriétés est un entier pour partager les conditions référencées en un ou plusieurs groupes, et la seconde est un booléen pour indiquer si une condition référencée est niée. Un exemple montre comment ConditionListType et ces deux propriétés supplémentaires fournissent une représentation

univoque d'un ensemble de conditions en DNF ou en CNF.

Supposons qu'on ait une PolicyRule qui agrège cinq PolicyCondition C1 à C5, avec les valeurs suivantes dans les propriétés des cinq associations PolicyConditionInPolicyRule :

C1 : GroupNumber = 1, ConditionNegated = FAUX
 C2 : GroupNumber = 1, ConditionNegated = VRAI
 C3 : GroupNumber = 1, ConditionNegated = FAUX
 C4 : GroupNumber = 2, ConditionNegated = FAUX
 C5 : GroupNumber = 2, ConditionNegated = FAUX

Si ConditionListType = DNF, alors la condition globale pour la PolicyRule est :

(C1 ET (NON C2) ET C3) OU (C4 ET C5)

D'un autre côté, si ConditionListType = CNF, la condition globale pour la PolicyRule est alors :

(C1 OU (NON C2) OU C3) ET (C4 OU C5)

Dans les deux cas, il y a une spécification non ambiguë de la condition globale essayée pour déterminer si il faut effectuer les actions associées à la PolicyRule.

La définition de classe est la suivante :

NOM : PolicyRule
 DESCRIPTION : classe centrale pour représenter la sémantique "Si condition alors action" associée à une règle de politique.
 DÉDUIT DE : Policy
 ABSTRAIT : FAUX
 PROPRIÉTÉS : Activé
 ConditionListType
 RuleUsage
 Priorité
 Obligatoire
 SequencedActions
 PolicyRoles

La classe PolicyRule est directement instanciable. Dans une mise en œuvre, diverses propriétés clé/identification DOIVENT être définies. Les clés pour une mise en œuvre native de CIM sont définies à l'Appendice A, paragraphe 13.1.2. Les clés pour une mise en œuvre LDAP seront définies dans la transposition LDAP du présent modèle d'informations [11].

6.3.1 Propriété "Enabled"

Cette propriété indique si une règle de politique est actuellement activée, d'un point de vue administratif. Son objet est de permettre à un administrateur de politique d'activer ou désactiver une règle de politique sans avoir à l'ajouter, ou la retirer du répertoire de politiques.

La propriété prend aussi en charge la valeur 'enabledForDebug'. Lorsque la propriété a cette valeur, l'entité qui évalue la ou les conditions de politique comprend qu'elle doit évaluer les conditions pour la règle de politique, mais pas d'effectuer les actions si les conditions s'évaluent à VRAI. Cette valeur sert de véhicule de débogage lors d'une tentative pour déterminer quelles politiques exécuterait un scénario particulier, sans effectuer d'action pour changer l'état durant le débogage.

La définition de la propriété est la suivante :

NOM : Enabled
 DESCRIPTION : énumération indiquant si une règle de politique est administrativement activée, administrativement désactivée, ou activée pour le mode débogage.
 SYNTAXE : uint16
 VALEURS : enabled(1), disabled(2), enabledForDebug(3)
 PAR DÉFAUT : VALEUR enabled(1)

6.3.2 Propriété "ConditionListType"

Cette propriété est utilisée pour spécifier si la liste des conditions de politique associée à cette règle de politique est en forme disjonctive normale (DNF, *disjunctive normal form*) ou en forme conjonctive normale (CNF, *conjunctive normal*

form). Si cette propriété n'est pas présente, le type de liste par défaut est DNF. La définition de la propriété est la suivante :

NOM : ConditionListType
 DESCRIPTION : indique si la liste des conditions de politique associée à cette règle de politique est en forme disjonctive normale (DNF) ou en forme conjonctive normale (CNF).
 SYNTAXE : uint16
 VALEURS : DNF(1), CNF(2)
 PAR DÉFAUT : VALEUR DNF(1)

6.3.3 Propriété "RuleUsage"

Cette propriété est une chaîne de forme libre qui recommande comment cette politique devrait être utilisée. La définition de la propriété est la suivante :

NOM : RuleUsage
 DESCRIPTION : cette propriété est utilisée pour donner des lignes directrices sur la façon dont cette politique devrait être utilisée.
 SYNTAXE : chaîne

6.3.4 Propriété "Priority"

Cette propriété donne un entier non négatif pour donner une priorité aux règles de politique les unes par rapport aux autres. Les plus fortes valeurs d'entier indiquent une plus forte priorité. Comme un des objets de cette propriété est de permettre que des règles de politique ad hoc spécifiques outrepassent temporairement des règles de politique établies, une instance qui a cette propriété établie a une priorité supérieure à celle de toutes les instances qui l'utilisent ou a la valeur par défaut de zéro.

L'établissement de priorités parmi les règles de politique donne un mécanisme de base pour résoudre les conflits de politique.

La définition de la propriété est la suivante :

NOM : Priority
 DESCRIPTION : entier non négatif pour donner la priorité à cette PolicyRule par rapport aux autres PolicyRule. Une plus grande valeur indique une plus forte priorité.
 SYNTAXE : uint16
 PAR DÉFAUT : VALEUR 0

6.3.5 Propriété "Mandatory"

Cette propriété indique si l'évaluation (et éventuellement l'exécution de l'action) d'une PolicyRule est obligatoire ou non. Ce concept est similaire à la capacité de marquer des paquets pour livraison ou pour élimination, sur la base du trafic réseau et de la charge de l'appareil.

L'évaluation d'une PolicyRule DOIT être tentée si la valeur de la propriété Mandatory est VRAI. Si la valeur de la propriété Mandatory d'une PolicyRule est FAUX, l'évaluation de la règle est alors "au mieux" et PEUT être ignorée.

La définition de la propriété est la suivante :

NOM : Mandatory
 DESCRIPTION : fanion qui indique que l'évaluation des PolicyCondition et l'exécution des PolicyAction (si la liste de conditions est évaluée à VRAI) est requise.
 SYNTAXE : booléen
 PAR DÉFAUT : VALEUR VRAI

6.3.6 Propriété "SequencedActions"

Cette propriété donne à un administrateur de politique un moyen de spécifier comment l'ordre des actions de politique associées à cette PolicyRule est à interpréter. Trois valeurs sont acceptées :

- o mandatory(1) : Faire les actions dans l'ordre indiqué, ou ne pas les faire du tout.
- o recommended(2) : Faire les actions dans l'ordre indiqué si on peut, mais si on ne peut pas les faire dans cet ordre, les faire dans un autre ordre si on peut.
- o dontCare(3) : Les faire – et ne pas se soucier de leur ordre.

Lorsque un rapport d'erreur/événement est adressé pour le cadre de politique, des codes convenables seront définis pour

rapporter qu'un ensemble d'actions n'ont pas pu être effectuées dans un ordre spécifié comme obligatoire (et donc n'ont pas été effectuées du tout), qu'un ensemble d'actions n'ont pas pu être effectuées dans un ordre recommandé (et de plus n'ont pas pu être effectuée dans n'importe quel ordre), ou qu'un ensemble d'actions n'ont pas pu être effectuées dans un ordre recommandé (mais ont été effectuées dans un ordre différent). La définition de la propriété est la suivante :

NOM : SequencedActions
 DESCRIPTION : énumération indiquant comment interpréter l'ordre des actions indiqué via l'agrégation PolicyActionInPolicyRule.
 SYNTAXE : uint16
 VALEURS : mandatory(1), recommended(2), dontCare(3)
 PAR DÉFAUT : VALEUR dontCare(3)

6.3.7 Propriété multi valeurs "PolicyRoles"

Cette propriété représente les rôles et combinaisons de rôles associés à une règle de politique. Chaque valeur représente une combinaison de rôles. Comme c'est une propriété multi valeurs, plus d'une combinaison de rôles peut être associée à une seule règle de politique. Chaque valeur est une chaîne de la forme :

```
<RoleName>[&&<RoleName>]*
```

où les noms de rôle individuels apparaissent en ordre alphabétique (selon la séquence de collationnement de l'UCS-2). La définition de la propriété est la suivante :

NOM : PolicyRoles
 DESCRIPTION : ensemble de chaînes représentant les rôles et combinaisons de rôles associés à une règle de politique. Chaque valeur représente une combinaison de rôles.
 SYNTAXE : chaîne

6.4 Classe abstraite "PolicyCondition"

L'objet d'une condition de politique est de déterminer si l'ensemble d'actions (agrégées dans la PolicyRule à laquelle s'applique la condition) devrait ou non être exécuté. Pour les besoins du modèle d'informations de cœur de politique, tout ce qui importe à propos d'une PolicyCondition individuelle est qu'elle s'évalue à VRAI ou FAUX. (Les PolicyCondition individuelles associées à une PolicyRule sont combinées pour former une expression composée en DNF ou en CNF, mais ceci est accompli via la propriété ConditionListType, exposée précédemment, et par les propriétés de l'agrégation PolicyConditionInPolicyRule, introduite plus haut et développée au paragraphe 7.6.) Une structure logique au sein d'une PolicyCondition individuelle peut aussi être introduite, mais cela devrait être fait dans une sous classe de PolicyCondition.

À cause de sa généralité, la classe PolicyCondition ne contient pas par elle-même de condition "réelle". Celle-ci seront représentées par les propriétés des sous classes spécifiques de domaine de PolicyCondition.

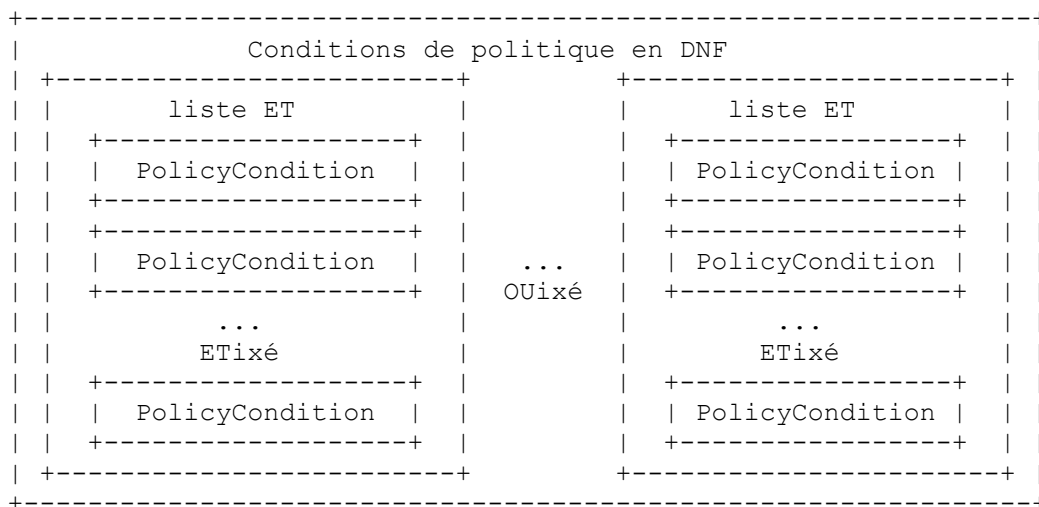


Figure 7 : Vue d'ensemble des conditions de politique en DNF

Cette figure illustre que lorsque les conditions de politique sont en DNF, il y a un ou plusieurs ensembles de conditions qui sont ETixées ensemble pour former des listes ET. Une liste ET s'évalue à VRAI si et seulement si toutes ses conditions constitutantes s'évaluent à VRAI. La condition globale s'évalue alors à VRAI si et seulement si au moins une de ses listes

ET constituante s'évalue à VRAI.

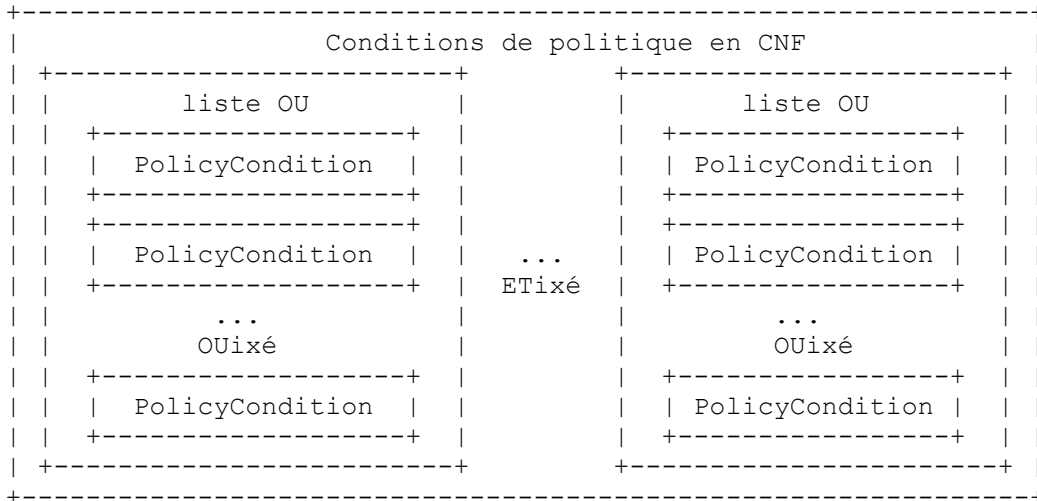


Figure 8 : Vue d'ensemble des conditions de politique en CNF

Dans cette figure, les conditions de politique sont en CNF. Par conséquent, il y a une ou plusieurs listes OU, dont chacune s'évalue à VRAI si et seulement si au moins une de ses conditions constituante s'évalue à VRAI. La condition globale s'évalue alors à VRAI si et seulement si TOUTES ses listes OU constituantes s'évaluent à VRAI.

La définition de classe de PolicyCondition est la suivante :

NOM : PolicyCondition

DESCRIPTION : classe représentant une condition de politique spécifique de règle ou réutilisable à évaluer en conjonction avec une règle de politique.

DÉDUIT DE : Policy

ABSTRAIT : VRAI

PROPRIÉTÉS : AUCUNE

Aucune propriété n'est définie pour cette classe car elle hérite toutes ses propriétés de Policy. La classe existe comme une super classe abstraite pour les conditions de politique spécifiques d'un domaine, définie en sous classes. Dans une mise en œuvre, diverses propriétés de clé/identification DOIVENT être définies pour la classe ou ses sous classes instanciables. Les clés pour une mise en œuvre native de CIM sont définies à l'Appendice A, paragraphe 13.2. Les clés pour une mise en œuvre LDAP seront définies dans la transposition LDAP de ce modèle d'informations [11].

Lors de l'identification et l'utilisation de la classe PolicyCondition, il est nécessaire de se souvenir qu'une condition peut être spécifique d'une règle ou réutilisable. Cela a été expliqué au paragraphe 5.1. La distinction entre les deux types de conditions de politique tient aux associations dans lesquelles une instance peut participer, et à la façon dont les différentes instances sont désignées. Conceptuellement, une condition de politique réutilisable réside dans un répertoire de politiques, et est nommée dans la portée de ce répertoire. D'un autre côté, une condition de politique spécifique d'une règle, comme son nom le suggère, est nommée dans la portée d'une seule règle de politique à laquelle elle se rapporte.

La distinction entre PolicyCondition spécifique de règle et réutilisable affecte la dénomination CIM, définie dans l'Appendice A, et la transposition LDAP [11].

6.5 Classe "PolicyTimePeriodCondition"

Cette classe donne le moyen de représenter les périodes durant lesquelles une règle de politique est valide, c'est-à-dire, active. Pendant toutes les périodes qui tombent en dehors de ces périodes, la règle de politique n'a pas d'effet. Une règle de politique est traitée comme valide à tout instant si elle ne spécifie pas une PolicyTimePeriodCondition.

Dans certains cas, un PDP peut avoir besoin d'effectuer certaines actions d'établissement/nettoyage lorsque une règle de politique devient active/inactive. Par exemple, les sessions qui ont été établies alors qu'une règle de politique était active peuvent devoir être supprimées lorsque la règle devient inactive. Cependant dans d'autres cas, de telles sessions pourraient rester en place : dans ce cas, l'effet de la désactivation de la règle de politique devrait seulement être d'empêcher l'établissement de nouvelles sessions. Les comportements d'établissement/suppression lors des transitions de période de

validité ne sont pas traités actuellement par le PCIM, et doivent être spécifiés dans des documents de "lignes directrices", ou via des sous classes de PolicyRule, PolicyTimePeriodCondition ou autres sous classes concrètes de Policy. Si de tels comportements ont besoin d'être sous le contrôle de l'administrateur de politique, un mécanisme pour permettre ce contrôle doit alors aussi être spécifié dans les sous classes.

PolicyTimePeriodCondition est défini comme une sous classe de PolicyCondition. C'est pour permettre l'inclusion de critères fondés sur l'heure dans les définitions de condition ET/OU pour une PolicyRule.

Les instances de cette classe peuvent aller jusqu'à cinq propriétés identifiant des périodes de temps à différents niveaux. Les valeurs de toutes les propriétés présentes dans une instance sont ETixées ensemble pour déterminer la ou les périodes de validité de l'instance. Par exemple, une instance avec une gamme globale de validité du 1^{er} janvier 2000 au 31 décembre 2000 ; un gabarit mensuel qui choisit mars et avril ; un gabarit de jour de la semaine qui choisit les vendredis ; et une plage d'heures du jour de 0800 à 1600 représenterait les périodes suivantes :

- vendredi 5 mars 2000, de 0800 à 1600;
- vendredi 12 mars 2000, de 0800 à 1600;
- vendredi 19 mars 2000, de 0800 à 1600;
- vendredi 26 mars 2000, de 0800 à 1600;
- vendredi 2 avril 2000, de 0800 à 1600;
- vendredi 9 avril 2000, de 0800 à 1600;
- vendredi 16 avril 2000, de 0800 à 1600;
- vendredi 23 avril 2000, de 0800 à 1600;
- vendredi 30 avril 2000, de 0800 à 1600.

Les propriétés qui ne sont pas présentes dans une instance de PolicyTimePeriodCondition sont traitées implicitement comme ayant leur valeur "toujours activée". Donc, dans l'exemple ci-dessus, le gabarit de jour du mois n'est pas présent, et la période de validité de l'instance inclut implicitement un gabarit de jour du mois qui choisit tous les jours du mois. Si on applique intégralement cette règle de la "propriété manquante", on voit qu'il y a une seconde façon d'indiquer qu'une règle de politique est toujours activée : en la faisant pointer sur une instance de PolicyTimePeriodCondition dont les seules propriétés sont ses propriétés de désignation.

La propriété LocalOrUtcTime indique si les périodes représentées dans les cinq autres propriétés en rapport avec le temps d'une instance de PolicyTimePeriodCondition sont à interpréter comme des heures locales pour la localisation où une règle de politique est appliquée, ou comme des temps UTC.

La définition de classe est la suivante .

NOM : PolicyTimePeriodCondition
 DESCRIPTION : Classe qui donne la capacité d'activer/désactiver une règle de politique selon un programme prédéterminé.
 DÉDUIT DE : PolicyCondition
 ABSTRAIT : FAUX
 PROPRIÉTÉS : TimePeriod
 MonthOfYearMask
 DayOfMonthMask
 DayOfWeekMask
 TimeOfDayMask
 LocalOrUtcTime

6.5.1 Propriété "TimePeriod"

Cette propriété identifie une gamme globale de dates et heures calendaires sur laquelle une règle de politique est valide. Elle réutilise le format d'une période explicite définie dans la RFC 2445 (référence [10]) : une chaîne représentant une date et heure de début, dans laquelle le caractère 'T' indique le début de la portion de temps, suivie par le caractère barre oblique '/', suivi par une chaîne similaire représentant une date et heure de fin. La première date indique le début de la gamme, tandis que la seconde date indique la fin. Donc, la seconde date et heure doit être postérieure à la première. Date/heure est exprimé comme une sous chaîne de la forme "aaaammjjThhmmss". Par exemple :

20000101T080000/20000131T120000

du 1 janvier 2000, 0800 au 31 janvier 2000, à midi

Il y a aussi deux cas particuliers dans lesquels une des chaîne date/heure est remplacée par une chaîne spéciale définie dans

la RFC 2445.

- o Si la première date/heure est remplacée par la chaîne "THISANDPRIOR", la propriété indique alors qu'une règle de politique est valide [de maintenant] jusqu'à la date/heure qui apparaît après le caractère '/'.
- o Si la seconde date/heure est remplacée par la chaîne "THISANDFUTURE", la propriété indique alors qu'une règle de politique devient valide à la date/heure qui apparaît avant le '/', et reste valide depuis ce point.

Noter que la RFC 2445 n'utilise pas ces deux chaînes en connexion avec des périodes explicites. Donc, le PCIM combine deux éléments de la RFC 2445 qui ne sont pas combinés dans la RFC elle-même.

La définition de la propriété est la suivante :

NOM : TimePeriod
 DESCRIPTION : gamme de dates calendaires sur lesquelles une règle de politique est valide.
 SYNTAXE : chaîne
 FORMAT : aaaammjjThhmmss/aaaammjjThhmmss, où la première date/heure peut être remplacée par la chaîne "THISANDPRIOR" ou la seconde date/heure peut être remplacée par la chaîne "THISANDFUTURE"

6.5.2 Propriété "MonthOfYearMask"

L'objet de cette propriété est de préciser la définition de la période de validité qui est définie par la propriété TimePeriod, en spécifiant explicitement les mois pendant lesquels la politique est valide. Ces propriétés fonctionnent ensemble, avec la TimePeriod utilisée pour spécifier la période globale durant laquelle la politique peut être valide, et le MonthOfYearMask utilisé pour prendre les mois spécifiques dans la période pendant lesquels la politique est valide.

Cette propriété est formatée comme une chaîne d'octets de taille 2, consistant en 12 bits identifiant les 12 mois de l'année, commençant à janvier et finissant à décembre, suivis par 4 bits qui sont toujours réglés à '0'. Pour chaque mois, la valeur '1' indique que la politique est valide pour ce mois, et la valeur '0' indique qu'elle n'est pas valide. La valeur X'08 30', par exemple, indique qu'une règle de politique n'est valide que les mois de mai, novembre, et décembre.

Voir au paragraphe 5.4 les détails de la façon dont CIM représente la chaîne d'octets d'une propriété d'une seule valeur telle que celle-ci. (Fondamentalement, CIM ajoute en tête une longueur de 4 octets à la chaîne d'octets.)

Si cette propriété est omise, la règle de politique est alors traitée comme valide pour tous les douze mois. La définition de la propriété est la suivante :

NOM : MonthOfYearMask
 DESCRIPTION : gabarit qui identifie les mois de l'année pendant lesquels une règle de politique est valide.
 SYNTAXE : chaîne d'octets
 FORMAT : X'hh h0'

6.5.3 Propriété "DayOfMonthMask"

L'objet de cette propriété est de préciser la définition de la période de validité qui est définie par la propriété TimePeriod, en spécifiant explicitement les jours du mois pendant lesquels la politique est valide. Ces propriétés fonctionnent ensemble, TimePeriod étant utilisé pour spécifier la période globale durant laquelle la politique peut être valide, et DayOfMonthMask étant utilisé pour retenir les jours spécifiques du mois au sein de cette période où la politique est valide.

Cette propriété est formatée comme une chaîne d'octets de taille 8, consistant en 31 bits qui identifient les jours du mois en comptant depuis le début, suivis par 31 bits de plus qui identifient les jours du mois en comptant depuis la fin, suivis par 2 bits qui sont toujours réglés à '0'. Pour chaque jour, la valeur '1' indique que la politique est valide pour ce jour, et la valeur '0' indique qu'elle n'est pas valide.

La valeur X'80 00 00 01 00 00 00 00', par exemple, indique qu'une règle de politique est valide le premier et le dernier jour du mois.

Pour les mois qui ont moins de 31 jours, les chiffres correspondant aux jours que le mois n'a pas (en comptant dans les deux directions) sont ignorés.

Le codage des 62 bits de poids fort dans la chaîne d'octets correspond à celui utilisé pour l'objet schedDay dans la DISMAN-SCHEDULE-MIB. Voir dans la référence [8] les détails de cet objet.

Voir au paragraphe 5.4 les détails de la façon dont CIM représente une chaîne d'octets de propriété d'une seule valeur telle

que celle-ci. (Fondamentalement, CIM ajoute en tête une longueur de 4 octets à la chaîne d'octets.)

La définition de la propriété est la suivante :

NOM : DayOfMonthMask
 DESCRIPTION : gabarit qui identifie les jours du mois pendant lesquels une règle de politique est valide.
 SYNTAXE : chaîne d'octets
 FORMAT : X'hh hh hh hh hh hh hh hh'

6.5.4 Propriété "DayOfWeekMask"

L'objet de cette propriété est de préciser la définition de la période de validité qui est définie par la propriété TimePeriod en spécifiant explicitement les jours de la semaine où la politique est valide. Ces propriétés fonctionnent ensemble, TimePeriod étant utilisé pour spécifier la période globale où la politique pourrait être valide, et DayOfWeekMask étant utilisé pour choisir les jours spécifiques de la semaine dans cette période où la politique est valide.

Cette propriété est formatée en une chaîne d'octets de taille 1, consistant en 7 bits qui identifient les 7 jours de la semaine, en commençant par dimanche et finissant par samedi, suivie par 1 bit qui est toujours réglé à '0'. Pour chaque jour de la semaine, la valeur '1' indique que la politique est valide ce jour, et la valeur '0' indique qu'elle n'est pas valide.

La valeur X'7C', par exemple, indique qu'une règle de politique est valide du lundi au vendredi.

Voir au paragraphe 5.4 les détails de la façon dont CIM représente la chaîne d'octets d'une propriété d'une seule valeur telle que celle-ci. (Fondamentalement, CIM ajoute en tête une longueur de 4 octets à la chaîne d'octets.)

La définition de la propriété est la suivante :

NOM : DayOfWeekMask
 DESCRIPTION : gabarit identifiant les jours de la semaine où une règle de politique est valide.
 SYNTAXE : chaîne d'octets
 FORMAT : B'bbbb bbb0'

6.5.5 Propriété "TimeOfDayMask"

L'objet de cette propriété est de préciser la définition de la période de validité qui est définie par la propriété TimePeriod en spécifiant explicitement une plage d'heures de la journée pendant lesquelles la politique est valide. Ces propriétés fonctionnent ensemble, avec la TimePeriod utilisée pour spécifier la période globale pendant laquelle la politique est valide, et le TimeOfDayMask utilisé pour prendre la gamme de périodes d'un certain jour dans cette période où la politique est valide.

Cette propriété est formatée dans le style de la RFC 2445 [10] : une chaîne horaire commençant par le caractère 'T', suivie par le caractère barre oblique '/', suivi par une seconde chaîne horaire. La première heure indique le début de la gamme, tandis que la seconde heure en indique la fin. Les heures sont exprimées par des sous chaînes de forme "Thhmmss".

La seconde sous chaîne identifie toujours une heure postérieure à celle de la première sous chaîne. Pour permettre cependant des gammes qui s'étendent au delà de minuit, la valeur de la seconde chaîne peut être inférieure à la valeur de la première sous chaîne. Donc, "T080000/T210000" identifie la plage de 0800 à 2100, tandis que "T210000/T080000" identifie la plage de 2100 à 0800 du jour suivant.

Lorsque une plage horaire s'étend au delà de minuit, elle inclut par définition des parties de deux jours successifs. Lorsque un de ces jours est aussi choisi par MonthOfYearMask, DayOfMonthMask, et/ou DayOfWeekMask, mais que l'autre jour ne l'est pas, la politique n'est alors active que durant la portion de la plage qui tombe sur le jour choisi. Par exemple, si la plage s'étend de 2100 à 0800, et si le gabarit de jours de la semaine a choisi lundi et mardi, la politique est alors active durant les trois intervalles suivants :

- du dimanche minuit au lundi 0800 ;
- du lundi 2100 au mardi 0800 ;
- du mardi 2100 au mardi 23:59:59.

La définition de la propriété est la suivante :

NOM : TimeOfDayMask
 DESCRIPTION : la plage horaire sur laquelle une règle de politique est valide. Si la seconde heure est antérieure à la

première, l'intervalle s'étend au delà de minuit.
 SYNTAXE : Chaîne
 FORMAT / Thhmmss/Thhmmss

6.5.6 Propriété "LocalOrUtcTime"

Cette propriété indique si les heures représentées dans la propriété TimePeriod et dans les diverses propriétés Mask représentent des heures locales ou des temps UTC. Il n'y a aucune disposition pour mélanger des heures locales et les temps UTC : les valeurs de cette propriété s'appliquent à toutes les autres propriétés en rapport avec l'heure.

La définition de la propriété est la suivante :

NOM : LocalOrUtcTime
 DESCRIPTION : indication de si les autres heures dans cette instance représentent des heures locales ou des temps UTC.
 SYNTAXE : uint16
 VALEURS : localTime(1), utcTime(2)
 PAR DÉFAUT : VALEUR utcTime(2)

6.6 Classe "VendorPolicyCondition"

L'objet de cette classe est de fournir un mécanisme général d'extension pour représenter les conditions de politique qui n'ont pas été modélisées avec des propriétés spécifiques. Les deux propriétés Constraint et ConstraintEncoding sont utilisées à la place pour définir le contenu et le format de la condition, comme expliqué ci-dessous.

Comme son nom le suggère, cette classe est destinée aux extensions spécifiques de fabricant au modèle d'informations de cœur de politique. Les extensions normalisées ne sont pas supposées utiliser cette classe.

La définition de classe est la suivante :

NOM : VendorPolicyCondition
 DESCRIPTION : classe qui définit un moyen enregistré de décrire une condition de politique.
 DÉDUIT DE : PolicyCondition
 ABSTRAIT : FAUX
 PROPRIÉTÉS : Constraint[]
 ConstraintEncoding

6.6.1 Propriété multi valeurs "Constraint"

Cette propriété fournit un mécanisme général d'extension pour représenter les conditions de politique qui n'ont pas été modélisées avec des propriétés spécifiques. Le format de la chaîne d'octets dans le dispositif n'est pas spécifié dans la présente définition. Il est déterminé par la valeur d'OID mémorisée dans la propriété ConstraintEncoding. Comme ConstraintEncoding est à une seule valeur, toutes les valeurs de Constraint partagent le même format et la même sémantique.

Voir au paragraphe 5.4 une description de la façon dont CIM code un dispositif de chaîne d'octets comme celui-ci.

Un point de décision de politique peut directement déterminer si il accepte les valeurs mémorisées dans une instance de Constraint en vérifiant la valeur d'OID de ConstraintEncoding par rapport à l'ensemble des OID qu'il reconnaît. L'action à effectuer pour le point de décision de politique au cas où il ne reconnaît pas le format de ces données pourrait elle-même être modélisée comme règle de politique, gouvernant le comportement du point de décision de politique.

La propriété est définie comme suit :

NOM : Constraint
 DESCRIPTION : mécanisme d'extension pour représenter les contraintes qui n'ont pas été modélisées comme propriétés spécifiques. Le format des valeurs est identifié par l'OID mémorisé dans la propriété ConstraintEncoding.
 SYNTAXE : chaîne d'octets

6.6.2 Propriété "ConstraintEncoding"

Cette propriété identifie le codage et la sémantique des valeurs de la propriété Constraint dans cette instance. La valeur de cette propriété est une seule chaîne, représentant un seul OID.

La propriété est définie comme suit :

NOM : ConstraintEncoding
 DESCRIPTION : OID codé comme chaîne, identifiant le format et la sémantique pour la propriété Constraint de cette instance. La valeur est une suite de chiffres décimaux séparés par des points (Par exemple, "1.2.100.200") représentant les arcs de l'OID. Les caractères dans la chaîne sont les caractères en UCS-2 qui correspondent au codage US ASCII des caractères numériques et du point.
 SYNTAXE : chaîne

6.7 Classe abstraite "PolicyAction"

L'objet d'une action de politique est d'exécuter une ou plusieurs opérations qui vont affecter le trafic du réseau et/ou des systèmes, des appareils, etc., afin de réaliser un état désiré. Ce (nouvel) état donne un ou plusieurs (nouveaux) comportements. Une action de politique change ordinairement la configuration d'un ou plusieurs éléments.

Une PolicyRule contient une ou plusieurs actions de politique. Un administrateur de politique peut assigner un ordre aux actions associées à une PolicyRule, les compléter par une indication de si l'ordre indiqué est obligatoire, recommandé, ou n'a pas de signification. L'ordre des actions associées à une PolicyRule est réalisé via une propriété dans l'agrégation PolicyActionInPolicyRule.

Les actions associées à une PolicyRule sont exécutées si et seulement si la ou les conditions globales de la PolicyRule s'évaluent à VRAI.

La définition de classe de PolicyAction est la suivante :

NOM : PolicyAction
 DESCRIPTION : classe représentant une action de politique spécifique d'une règle ou réutilisable à effectuer si la condition pour une règle de politique s'évalue à VRAI.
 DÉDUIT DE : Policy
 ABSTRAIT : VRAI
 PROPRIÉTÉS : AUCUNE

Aucune propriété n'est définie pour cette classe car elle hérite toutes ses propriétés de Policy. La classe existe comme super classe abstraite pour des actions de politique spécifiques d'un domaine, définies dans des sous classes. Dans une mise en œuvre, diverses propriétés de clé/identification DOIVENT être définies pour la classe ou ses sous classes instanciables. Les clés pour une mise en œuvre native de CIM sont définies dans l'Appendice A, au paragraphe 13.3. Les clés pour une mise en œuvre LDAP seront définies dans la transposition LDAP du présent modèle d'informations [11].

Pour identifier et utiliser la classe PolicyAction, il est nécessaire de se souvenir qu'une action peut être spécifique d'une règle ou réutilisable. Cela a été expliqué au paragraphe 5.1. La distinction entre les deux types d'actions de politique tient aux associations auxquelles peut participer une instance, et à la façon dont les différentes instances sont nommées. Conceptuellement, une action de politique réutilisable réside dans un répertoire de politiques, et est désignée au sein de la portée de ce répertoire. D'un autre côté, une action de politique spécifique d'une règle est désignée au sein de la portée de la seule règle de politique à laquelle elle se rapporte.

La distinction entre PolicyAction spécifique d'une règle et réutilisable affecte la désignation CIM, définie dans l'Appendice A, et la transposition LDAP [11].

6.8 Classe "VendorPolicyAction"

L'objet de cette classe est de fournir un mécanisme général d'extension pour représenter des actions de politique qui n'ont pas été modélisées avec des propriétés spécifiques. Les deux propriétés ActionData et ActionEncoding sont plutôt utilisées pour définir le contenu et le format de l'action, comme expliqué ci-dessous.

Comme son nom le suggère, cette classe est destinée aux extensions spécifiques de fabricant au modèle d'informations de cœur de politique. Les extensions normalisées ne sont pas supposées utiliser cette classe.

La définition de classe est la suivante :

NOM : VendorPolicyAction
 DESCRIPTION : classe qui définit un moyen enregistré de décrire une action de politique.
 DÉDUIT DE : PolicyAction
 ABSTRAIT : FAUX
 PROPRIÉTÉS : ActionData[]
 ActionEncoding

6.8.1 Propriété multi valeurs "ActionData"

Cette propriété fournit un mécanisme général d'extension pour représenter des actions de politique qui n'ont pas été modélisées avec des propriétés spécifiques. Le format de la chaîne d'octets dans le dispositif n'est pas spécifié dans cette définition. Il est déterminé par la valeur de l'OID mémorisée dans la propriété ActionEncoding. Comme ActionEncoding est à une seule valeur, toutes les valeurs de ActionData partagent le même format et la même sémantique. Voir au paragraphe 5.4 l'explication de la façon dont CIM code un dispositif de chaîne d'octets comme celui-ci.

Un point de décision de politique peut directement déterminer si il accepte les valeurs mémorisées dans une instance de ActionData en vérifiant la valeur de l'OID de ActionEncoding par rapport à l'ensemble d'OID qu'il reconnaît. L'action que doit effectuer le point de décision de politique au cas où il ne reconnaît pas le format de ces données pourrait elle-même être modélisée comme une règle de politique, gouvernant le comportement du point de décision de politique.

La propriété est définie comme suit :

NOM : ActionData
 DESCRIPTION : mécanisme d'extension pour représenter les actions qui n'ont pas été modélisées comme propriétés spécifiques. Le format des valeurs est identifié par l'OID mémorisé dans la propriété ActionEncoding.
 SYNTAXE : chaîne d'octets

6.8.2 Propriété "ActionEncoding"

Cette propriété identifie le codage et la sémantique des valeurs de la propriété ActionData dans cette instance. La valeur de cette propriété est une seule chaîne, représentant un seul OID.

La propriété est définie comme suit :

NOM : ActionEncoding
 DESCRIPTION : OID codé comme chaîne, identifiant le format et la sémantique de la propriété ActionData de cette instance. La valeur est une séquence de chiffres décimaux séparés par des points (par exemple, "1.2.100.200") représentant les arcs de l'OID. Les caractères de la chaîne sont les caractères UCS-2 correspondant aux codages en US ASCII des caractères numériques et du point.
 SYNTAXE : chaîne

6.9 Classe "PolicyRepository"

La définition de classe de PolicyRepository est la suivante :

NOM : PolicyRepository
 DESCRIPTION : classe représentant un conteneur défini administrativement pour des informations réutilisables en rapport avec la politique. Cette classe n'introduit aucune propriété supplémentaire au delà de celles de sa super classe AdminDomain. Elle participe cependant à un certain nombre d'associations uniques.
 DÉDUIT DE : AdminDomain
 ABSTRAIT : FAUX

7. Définitions d'association et d'agrégation

Les deux premiers paragraphes de cette section introduisent les associations et agrégations telles qu'elles sont utilisées dans CIM. Les paragraphes suivants présentent les définitions de classe pour les associations et agrégations qui font partie du modèle d'informations de cœur de politique.

7.1 Associations

Une association est une construction CIM qui représente une relation entre deux (ou théoriquement plus) objets. Elle est modélisée comme une classe contenant normalement deux références d'objet. Les associations peuvent être définies entre des classes sans affecter aucune des classes concernées. C'est-à-dire que l'ajout d'une association n'affecte pas l'interface des classes concernées.

7.2 Agrégations

Une agrégation est une forme forte d'association, qui représente habituellement une relation de "partie entière" ou de "collection". Par exemple, CIM utilise une agrégation pour représenter la relation de contenance entre un système et les composants qui constituent le système. Que l'agrégation ait une relation de "partie entière" implique souvent, mais pas nécessairement, que les objets agrégés aient des interdépendances mutuelles.

7.3 Agrégation abstraite "PolicyComponent"

Cette agrégation abstraite définit deux références d'objet qui seront outrepassées dans chacune des cinq sous classes, pour devenir des références aux classes de politique concrètes PolicyGroup, PolicyRule, PolicyCondition, PolicyAction, et PolicyTimePeriodCondition. La valeur de la super classe abstraite est de porter que toutes les cinq sous classes ont la même sémantique de "partie entière", et de faciliter l'interrogation de localisation de tous les "composants" d'un PolicyGroup ou d'une PolicyRule.

La définition de classe pour l'agrégation est la suivante :

NOM : PolicyComponent
 DESCRIPTION : agrégation générique utilisée pour établir un 'élément de' relation entre les sous classes de Policy. Par exemple, l'agrégation PolicyConditionInPolicyRule définit que PolicyConditions fait partie d'une PolicyRule.
 ABSTRACT : VRAI
 PROPRIÉTÉS : GroupComponent[ref Policy[0..n]]
 PartComponent[ref Policy[0..n]]

7.4 Agrégation "PolicyGroupInPolicyGroup"

L'agrégation PolicyGroupInPolicyGroup permet d'incorporer des groupes de politique. Ceci est critique pour l'adaptabilité et la gérabilité, car elle permet de construire pour des besoins administratifs des politiques complexes à partir de plusieurs politiques plus simples. Par exemple, un groupe de politique qui représente des politiques pour les USA pourrait avoir des groupes de politique incorporés pour l'est et l'ouest des États-Unis.

Un PolicyGroup peut agréger d'autres PolicyGroup via cette agrégation, ou il peut agréger des PolicyRule via l'agrégation PolicyRuleInPolicyGroup. Noter qu'on suppose que cette agrégation est utilisée pour former des graphes acycliques dirigés et NON des structures d'anneau. La définition de classe pour l'agrégation est la suivante :

NOM : PolicyGroupInPolicyGroup
 DESCRIPTION : classe représentant l'agrégation de PolicyGroup par un PolicyGroup de niveau supérieur.
 DÉDUIT DE : PolicyComponent
 ABSTRACT : FAUX
 PROPRIÉTÉS : GroupComponent[ref PolicyGroup[0..n]]
 PartComponent[ref PolicyGroup[0..n]]

7.4.1 Référence "GroupComponent"

Cette propriété est héritée de PolicyComponent, et est outrepassée pour devenir une référence d'objet à un PolicyGroup qui contient un ou plusieurs autres PolicyGroup. Noter que pour toute instance seule de la classe d'agrégation PolicyGroupInPolicyGroup, cette propriété (comme toutes les propriétés Reference) est d'une seule valeur. La cardinalité [0..n] indique qu'il peut y avoir 0, 1, ou plus d'un PolicyGroup qui contient un certain PolicyGroup.

7.4.2 Référence "PartComponent"

Cette propriété est héritée de PolicyComponent, et est outrepassée pour devenir une référence d'objet à un PolicyGroup contenu par un ou plusieurs autres PolicyGroup. Noter que pour toute instance seule de la classe d'agrégation PolicyGroupInPolicyGroup, cette propriété (comme toutes les propriétés Reference) est d'une seule valeur. La cardinalité [0..n] indique qu'un certain PolicyGroup peut contenir 0, 1, ou plus d'un autre PolicyGroup.

7.5 Agrégation "PolicyRuleInPolicyGroup"

Un groupe de politique peut agréger une ou plusieurs règles de politique, via l'agrégation PolicyRuleInPolicyGroup. Grouper des règles de politique dans un groupe de politiques est là encore pour des besoins administratifs ; une règle de politique peut aussi être utilisée par elle-même, sans appartenir à un groupe de politiques.

Un PolicyGroup peut agréger des PolicyRule via cette agrégation, ou il peut agréger d'autres PolicyGroup via l'agrégation PolicyGroupInPolicyGroup.

La définition de classe pour l'agrégation est la suivante :

```
NOM : PolicyRuleInPolicyGroup
DESCRIPTION : classe représentant l'agrégation de PolicyRule par un PolicyGroup.
DÉDUIT DE : PolicyComponent
ABSTRAIT : FAUX
PROPRIÉTÉS : GroupComponent[ref PolicyGroup[0..n]]
              PartComponent[ref PolicyRule[0..n]]
```

7.5.1 Référence "GroupComponent"

Cette propriété est héritée de PolicyComponent, et est outrepassée pour devenir une référence d'objet à un PolicyGroup qui contient une ou plusieurs PolicyRule. Noter que pour toute instance seule de la classe d'agrégation PolicyRuleInPolicyGroup, cette propriété (comme toutes les propriétés Reference) est à une seule valeur. La cardinalité [0..n] indique qu'il peut y avoir 0, 1, ou plus d'un PolicyGroup qui contient une certaine PolicyRule.

7.5.2 Référence "PartComponent"

Cette propriété est héritée de PolicyComponent, et est outrepassée pour devenir une référence d'objet à une PolicyRule contenue par un ou plusieurs PolicyGroup. Noter que pour toute instance seule de la classe d'agrégation PolicyRuleInPolicyGroup, cette propriété (comme toutes les propriétés Reference) est à une seule valeur. La cardinalité [0..n] indique qu'un certain PolicyGroup peut contenir 0, 1, ou plus d'une PolicyRule.

7.6 Agrégation "PolicyConditionInPolicyRule"

Une règle de politique agrège zéro, une ou plusieurs instances de la classe PolicyCondition, via l'association PolicyConditionInPolicyRule. Une règle de politique qui agrège zéro condition de politique doit indiquer dans sa définition de classe ce qui "déclanche" la réalisation de ses actions. En bref, elle doit décrire ses PolicyCondition implicites, car aucune n'est explicitement associée. Par exemple, il pourrait y avoir une sous-classe de PolicyRule nommée "HttpPolicyRule", où la définition de classe suppose que la condition "Si trafic HTTP" est vraie avant que les actions de la règle ne soient effectuées. Il n'est pas besoin de formaliser et instancier cette condition, car elle est évidente dans la sémantique de la PolicyRule.

Les conditions agrégées par une règle de politique sont groupées selon deux niveaux de listes : soit un ensemble OUIxé d'ensembles ETixés de conditions (DNF, par défaut) soit un ensemble ETixé d'ensembles OUIxés de conditions (CNF). Les conditions individuelles dans ces listes peuvent être niées. La propriété ConditionListType (dans PolicyRule) spécifie lequel de ces deux schémas de groupage s'applique à une PolicyRule particulière. Les conditions sont utilisées pour

déterminer si il faut effectuer les actions associées à la PolicyRule.

Une ou plusieurs périodes horaires de politique peuvent figurer parmi les conditions associées à une règle de politique via l'association PolicyConditionInPolicyRule. Dans ce cas, les périodes sont simplement des conditions supplémentaires à évaluer avec toutes les autres conditions spécifiées pour la règle.

La définition de classe pour l'agrégation est la suivante :

NOM : PolicyConditionInPolicyRule
 DESCRIPTION : classe représentant l'agrégation de PolicyCondition par une PolicyRule.
 DÉDUIT DE : PolicyComponent
 ABSTRAIT : FAUX
 PROPRIÉTÉS : GroupComponent[ref PolicyRule[0..n]]
 PartComponent[ref PolicyCondition[0..n]]
 GroupNumber
 ConditionNegated

7.6.1 Référence "GroupComponent"

Cette propriété est héritée de PolicyComponent, et est outrepassée pour devenir une référence d'objet à une PolicyRule qui contient une ou plusieurs PolicyCondition. Noter que pour toute instance seule de la classe d'agrégation PolicyConditionInPolicyRule, cette propriété (comme toutes les propriétés Reference) est d'une seule valeur. La cardinalité [0..n] indique qu'il peut y avoir 0, 1, ou plus d'une PolicyRule qui contient une certaine PolicyCondition.

7.6.2 Référence "PartComponent"

Cette propriété est héritée de PolicyComponent, et est outrepassée pour devenir une référence d'objet à une PolicyCondition contenue par une ou plusieurs PolicyRule. Noter que pour toute instance seule de la classe d'agrégation PolicyConditionInPolicyRule, cette propriété (comme toutes les propriétés Reference) est d'une seule valeur. La cardinalité [0..n] indique qu'une certaine PolicyRule peut contenir 0, 1, ou plus d'une PolicyCondition.

7.6.3 Propriété "GroupNumber"

Cette propriété contient un entier qui identifie le groupe auquel la condition référencée par la propriété PartComponent est allouée pour former l'expression conditionnelle globale pour la règle de politique identifiée par la référence de GroupComponent.

La propriété est définie comme suit :

NOM : GroupNumber
 DESCRIPTION : Entier non signé indiquant le groupe auquel la condition identifiée par la propriété PartComponent est à allouer.
 SYNTAXE : uint16
 PAR DÉFAUT : 0

7.6.4 Propriété "ConditionNegated"

Cette propriété est un booléen, qui indique si la condition référencée par la propriété PartComponent est niée dans la formation de l'expression conditionnelle globale pour la règle de politique identifiée par la référence GroupComponent.

La propriété est définie comme suit :

NOM : ConditionNegated
 DESCRIPTION : indication de si la condition identifiée par la propriété PartComponent est niée. (VRAI indique que la condition est niée, FAUX indique qu'elle n'est pas niée.)
 SYNTAXE : booléen
 PAR DÉFAUT : FAUX

7.7 Agrégation "PolicyRuleValidityPeriod"

Une relation différente (de PolicyConditionInPolicyRule) entre une règle de politique et une période de politique est représentée par l'agrégation PolicyRuleValidityPeriod. Cette dernière décrit l'activation et la désactivation programmées de la règle de politique.

Si une règle de politique est associée à plusieurs périodes de politique via cette association, la règle est alors active si au moins une des périodes indique qu'elle est active. (En d'autres termes, les périodes sont OUixées pour déterminer si la règle est active.) Une période de politique peut être agrégée par plusieurs règles de politique. Une règle qui ne pointe pas sur une période de politique via cette agrégation est, du point de vue de la programmation, toujours active. Elle peut, cependant, être inactive pour d'autres raisons.

Les périodes de temps sont un concept général qui peut être utilisé dans d'autres applications. Cependant, elles sont mentionnées explicitement ici dans la présente spécification car elles sont fréquemment utilisées dans les applications de politique.

La définition de classe pour l'agrégation est la suivante :

NOM : PolicyRuleValidityPeriod
 DESCRIPTION : classe représentant l'agrégation de PolicyTimePeriodConditions par une PolicyRule.
 DÉDUIT DE : PolicyComponent
 ABSTRAIT : FAUX
 PROPRIÉTÉS : GroupComponent[ref PolicyRule[0..n]]
 PartComponent[ref PolicyTimePeriodCondition[0..n]]

7.7.1 Référence "GroupComponent"

Cette propriété est héritée de PolicyComponent, et est outrepassées pour devenir une référence d'objet à une PolicyRule qui contient une ou plusieurs PolicyTimePeriodCondition. Noter que pour toute instance seule de la classe d'agrégation PolicyRuleValidityPeriod, cette propriété (comme toutes les propriétés Reference) est d'une seule valeur. La cardinalité [0..n] indique qu'il peut y avoir 0, 1, ou plus d'une PolicyRule qui contient une certaine PolicyTimePeriodCondition.

7.7.2 Référence "PartComponent"

Cette propriété est héritée de PolicyComponent, et est outrepassée pour devenir une référence d'objet pour une PolicyTimePeriodCondition contenue par une ou plusieurs PolicyRule. Noter que pour toute instance seule de la classe d'agrégation PolicyRuleValidityPeriod, cette propriété (comme toutes les propriétés Reference) est à une seule valeur. La cardinalité [0..n] indique qu'une certaine PolicyRule peut contenir 0, 1, ou plus d'une PolicyTimePeriodCondition.

7.8 Agrégation "PolicyActionInPolicyRule"

Une règle de politique peut agréger zéro, une ou plusieurs actions de politique. Une règle de politique qui agrège zéro action de politique doit indiquer dans sa définition de classe quelles actions sont prises lorsque les conditions de la règle s'évaluent à VRAI. En bref, elle doit décrire ses PolicyAction implicites, car aucune n'est explicitement associée. Par exemple, il peut y avoir une sous-classe de PolicyRule représentant un abandon Diffserv absolu, où la sous-classe elle-même indique l'action à effectuer. Il n'est pas nécessaire de formaliser et instancier cette action, car elle est évidente dans la sémantique de la PolicyRule.

Les actions associées à une PolicyRule peuvent recevoir un ordre obligatoire, un ordre recommandé, ou pas d'ordre du tout. Pour les actions représentées comme des objets séparés, l'agrégation PolicyActionInPolicyRule peut être utilisée pour exprimer un ordre.

Cette agrégation n'indique pas si un ordre d'action spécifié est requis, recommandé, ou n'a pas de signification ; la propriété SequencedActions dans l'instance d'agrégation de PolicyRule fournit cette indication.

La définition de classe pour l'agrégation est la suivante :

NOM : PolicyActionInPolicyRule
 DESCRIPTION : classe représentant l'agrégation de PolicyAction par une PolicyCondition.
 DÉDUIT DE : PolicyComponent
 ABSTRAIT : FAUX
 PROPRIÉTÉS : GroupComponent[ref PolicyRule[0..n]]
 PartComponent[ref PolicyAction[0..n]]
 ActionOrder

7.8.1 Référence "GroupComponent"

Cette propriété est héritée de PolicyComponent, et est outrepassée pour devenir une référence d'objet pour une PolicyRule qui contient une ou plusieurs PolicyAction. Noter que pour toute instance seule de la classe d'agrégation PolicyActionInPolicyRule, cette propriété (comme toutes les propriétés Reference) est d'une seule valeur. La cardinalité [0..n] indique qu'il peut y avoir 0, 1, ou plus d'une PolicyRule qui contient une certaine PolicyAction.

7.8.2 Référence "PartComponent"

Cette propriété est héritée de PolicyComponent, et est outrepassée pour devenir une référence d'objet pour une PolicyAction contenue par une ou plusieurs PolicyRule. Noter que pour toute instance seule de la classe d'agrégation PolicyActionInPolicyRule, cette propriété (comme toutes les propriétés Reference) est d'une seule valeur. La cardinalité [0..n] indique qu'une certaine PolicyRule peut contenir 0, 1, ou plus d'une PolicyAction.

7.8.3 Propriété "ActionOrder"

Cette propriété donne un entier non signé 'n' qui indique la position relative d'une action dans la séquence des actions associées à une règle de politique. Lorsque 'n' est un entier positif, il indique une place dans la séquence des actions à effectuer, les plus petits entiers indiquant des positions antérieures dans la séquence. La valeur spéciale '0' indique "ne pas tenir compte". Si deux actions ou plus ont le même numéro de séquence différent de zéro, elles peuvent être effectuées dans n'importe quel ordre, mais elles doivent toutes être effectuées à la place appropriée dans la séquence d'actions globale.

Une série d'exemples va éclaircir la question de l'ordre des actions :

- o Si toutes les actions ont le même numéro de séquence, sans considération de si il est '0' ou non, tout ordre est acceptable.
- o Les valeurs
 - 1 : ACTION A
 - 2 : ACTION B
 - 1 : ACTION C
 - 3 : ACTION D

indiquent deux ordres acceptables : A,C,B,D ou C,A,B,D, car A et C peuvent être effectuées dans l'un et l'autre ordre, mais seulement à la position '1'.

- o Les valeurs
 - 0 : ACTION A
 - 2 : ACTION B
 - 3 : ACTION C
 - 3 : ACTION D

exigent que B,C, et D surviennent soit comme B,C,D soit comme B,D,C. L'action A peut apparaître à tout moment par rapport à B,C, et D. Donc, le jeu complet des ordres acceptables est : A,B,C,D ; B,A,C,D ; B,C,A,D ; B,C,D,A ; A,B,D,C ; B,A,D,C ; B,D,A,C ; B,D,C,A.

Noter que les numéros de séquence non zéro n'ont pas besoin de commencer par '1', et ils n'ont pas besoin d'être consécutifs. Tout ce qui importe est leur magnitude relative.

La propriété est définie comme suit :

NOM : ActionOrder
 DESCRIPTION : entier non signé indiquant la position relative d'une action dans la séquence des actions agrégées par une règle de politique.
 SYNTAXE : uint16

7.9 Association abstraite "PolicyInSystem"

Cette association abstraite hérite de deux références d'objet d'une classe d'association CIM de niveau supérieur, Dependency. Elle outrepassé ces références d'objet pour faire référence aux instances des classes System et Policy. Les sous-classes de PolicyInSystem outrepassent alors encore une fois ces références d'objet pour les référencer à des classes concrètes de politique.

La valeur de la super classe abstraite est pour porter que toutes les sous classes ont la même sémantique "dependency", et pour faciliter l'interrogation de localisation de toutes les "dépendances" de politique sur un système. Ces dépendances se rapportent à la portée ou à l'hébergement de la politique.

La définition de classe pour l'association est la suivante :

NOM : PolicyInSystem
 DESCRIPTION : association générique utilisée pour établir des relations de dépendance entre les politiques et les systèmes qui les hébergent.
 DÉDUIT DE : Dependency
 ABSTRAIT : VRAI
 PROPRIÉTÉS : Antecedent[ref System[0..1]]
 Dependent[ref Policy[0..n]]

7.10. Association faible "PolicyGroupInSystem"

Cette association lie un PolicyGroup au système dans la portée duquel le PolicyGroup est défini.

La définition de classe pour l'association est la suivante :

NOM : PolicyGroupInSystem
 DESCRIPTION : classe représentant le fait qu'un PolicyGroup est défini dans la portée d'un système.
 DÉDUIT DE : PolicyInSystem
 ABSTRAIT : FAUX
 PROPRIÉTÉS : Antecedent[ref System[1..1]]
 Dependent[ref PolicyGroup[weak]]

7.10.1 Référence "Antecedent"

Cette propriété est héritée de PolicyInSystem, et est outrepassée pour restreindre sa cardinalité à [1..1]. Elle sert de référence d'objet à un système qui fournit une portée pour un ou plusieurs PolicyGroup. Comme c'est une association faible, la cardinalité pour cette référence d'objet est toujours 1, c'est-à-dire qu'un PolicyGroup est toujours défini au sein de la portée d'exactly un système.

7.10.2 Référence "Dependent"

Cette propriété est héritée de PolicyInSystem, et est outrepassée pour devenir une référence d'objet à un PolicyGroup défini au sein de la portée d'un système. Noter que pour toute instance seule de la classe d'association PolicyGroupInSystem, cette propriété (comme toutes les propriétés Reference) est à une seule valeur. La cardinalité [0..n] indique qu'un certain système peut avoir 0, 1, ou plus d'un PolicyGroup défini au sein de sa portée.

7.11 Association faible "PolicyRuleInSystem"

Sans considération de son appartenance à un PolicyGroup (ou à plusieurs PolicyGroup) une PolicyRule est elle-même définie au sein de la portée d'un système. Cette association lie une PolicyRule au système dont la PolicyRule est définie dans sa portée.

La définition de classe pour l'association est la suivante :

NOM : PolicyRuleInSystem
 DESCRIPTION ! classe représentant le fait qu'une PolicyRule est définie dans la portée d'un système.
 DÉDUIT DE : PolicyInSystem
 ABSTRAIT : FAUX
 PROPRIÉTÉS : Antecedent[ref System[1..1]]
 Dependent[ref PolicyRule[weak]]

7.11.1 Référence "Antecedent"

Cette propriété est héritée de PolicyInSystem, et est outrepassée pour restreindre sa cardinalité à [1..1]. Elle sert de référence d'objet à un système qui fournit une portée pour une ou plusieurs PolicyRule. Comme c'est une association faible, la cardinalité pour cette référence d'objet est toujours 1, c'est-à-dire qu'une PolicyRule est toujours définie dans la portée d'exactly un système.

7.11.2 Référence "Dependent"

Cette propriété est héritée de PolicyInSystem, et est outrepassée pour devenir une référence d'objet à une PolicyRule définie dans la portée d'un système. Noter que pour toute instance seule de la classe d'association PolicyRuleInSystem, cette propriété (comme toutes les propriétés Reference) est à une seule valeur. La cardinalité [0..n] indique qu'un certain système peut avoir 0, 1, ou plus d'une PolicyRule définie dans sa portée.

7.12 Association "PolicyConditionInPolicyRepository"

Une condition de politique réutilisable se rapporte toujours à un seul PolicyRepository, via l'association PolicyConditionInPolicyRepository. Ceci n'est pas vrai cependant pour toutes les PolicyCondition. Une instance de PolicyCondition qui représente une condition spécifique d'une règle ne se rapporte à aucun répertoire de politique via cette association.

La définition de classe pour l'association est la suivante :

NOM : PolicyConditionInPolicyRepository
 DESCRIPTION : classe représentant l'inclusion d'une PolicyCondition réutilisable dans un PolicyRepository.
 DÉDUIT DE : PolicyInSystem
 ABSTRACT : FAUX
 PROPRIÉTÉS : Antecedent[ref PolicyRepository[0..1]]
 Dependent[ref PolicyCondition[0..n]]

7.12.1 Référence "Antecedent"

Cette propriété est héritée de PolicyInSystem, et est outrepassée pour devenir une référence d'objet à un PolicyRepository contenant une ou plusieurs PolicyCondition. Une PolicyCondition réutilisable se rapporte toujours à exactement un PolicyRepository via l'association PolicyConditionInPolicyRepository. La cardinalité [0..1] pour cette propriété couvre les deux types de PolicyCondition : 0 pour une PolicyCondition spécifique d'une règle, 1 pour une réutilisable.

7.12.2 Référence "Dependent"

Cette propriété est héritée de PolicyInSystem, et est outrepassée pour devenir une référence d'objet à une PolicyCondition incluse dans un PolicyRepository. Noter que pour toute instance seule de la classe d'association PolicyConditionInPolicyRepository, cette propriété (comme toutes les propriétés Reference) est à une seule valeur. La cardinalité [0..n] indique qu'un certain PolicyRepository peut contenir 0, 1, ou plus d'une PolicyCondition.

7.13 Association "PolicyActionInPolicyRepository"

Une action de politique réutilisable est toujours en rapport avec un seul PolicyRepository, via l'association PolicyActionInPolicyRepository. Ceci n'est cependant pas vrai pour toutes les PolicyAction. Une instance de PolicyAction qui représente une action spécifique d'une règle ne se rapporte à aucun répertoire de politique via cette association.

La définition de classe pour l'association est la suivante :

NOM : PolicyActionInPolicyRepository
 DESCRIPTION : classe représentant l'inclusion d'une PolicyAction réutilisable dans un PolicyRepository.
 DÉDUIT DE : PolicyInSystem
 ABSTRACT : FAUX
 PROPRIÉTÉS : Antecedent[ref PolicyRepository[0..1]]
 Dependent[ref PolicyAction[0..n]]

7.13.1 Référence "Antecedent"

Cette propriété est héritée de PolicyInSystem, et est outrepassée pour devenir une référence d'objet à un PolicyRepository contenant une ou plusieurs PolicyAction. Une PolicyAction réutilisable se rapporte toujours à exactement un PolicyRepository via l'association PolicyActionInPolicyRepository. La cardinalité [0..1] pour cette propriété couvre les deux types de PolicyAction : 0 pour une PolicyAction spécifique d'une règle, 1 pour une réutilisable.

7.13.2 Référence "Dependent"

Cette propriété est héritée de PolicyInSystem, et est outrepassée pour devenir une référence d'objet à une PolicyAction incluse dans un PolicyRepository. Noter que pour toute instance seule de la classe d'association PolicyActionInPolicyRepository, cette propriété (comme toutes les propriétés Reference) est à une seule valeur. La cardinalité [0..n] indique qu'un certain PolicyRepository peut contenir 0, 1, ou plus d'une PolicyAction.

7.14 Agrégation "PolicyRepositoryInPolicyRepository"

L'agrégation PolicyRepositoryInPolicyRepository permet d'incorporer les répertoires de politique. Ceci découle de l'association CIM de niveau supérieur CIM_SystemComponent, qui décrit que les systèmes contiennent d'autres ManagedSystemElement. Cette superclasse ne pourrait pas être utilisée pour les autres agrégations de politique, car les politiques ne sont pas des ManagedSystemElement, mais des ManagedElement. Noter que il est supposé que cette agrégation est utilisée pour former des graphes dirigés acycliques et NON des structures d'anneau.

La définition de classe pour l'agrégation est la suivante :

NOM : PolicyRepositoryInPolicyRepository
 DESCRIPTION : classe représentant l'agrégation de plusieurs PolicyRepository par un PolicyRepository de niveau supérieur.
 DÉDUIT DE : SystemComponent
 ABSTRACT : FAUX
 PROPRIÉTÉS : GroupComponent[ref PolicyRepository[0..n]]
 PartComponent[ref PolicyRepository[0..n]]

7.14.1 Référence "GroupComponent"

Cette propriété est héritée de la classe CIM SystemComponent, et est outrepassée pour devenir une référence d'objet à un PolicyRepository qui contient un ou plusieurs autres PolicyRepository. Noter que pour toute instance seule de la classe d'agrégation PolicyRepositoryInPolicyRepository, cette propriété (comme toutes les propriétés Reference) est à une seule valeur. La cardinalité [0..n] indique qu'il peut y avoir 0, 1, ou plus d'un PolicyRepository qui contient tout PolicyRepository donné.

7.14.2 Référence "PartComponent"

Cette propriété est héritée de la classe CIM SystemComponent, et est outrepassée pour devenir une référence d'objet à un PolicyRepository contenu par un ou plusieurs autres PolicyRepository. Noter que pour toute instance seule de la classe d'agrégation PolicyRepositoryInPolicyRepository, cette propriété (comme toutes les propriétés Reference) est à une seule valeur. La cardinalité [0..n] indique qu'un certain PolicyRepository peut contenir 0, 1, ou plus d'un autre PolicyRepository.

8. Propriété intellectuelle

L'IETF ne prend pas position sur la validité et la portée de tout droit de propriété intellectuelle ou autres droits qui pourraient être revendiqués au titre de la mise en œuvre ou l'utilisation de la technologie décrite dans le présent document ou sur la mesure dans laquelle toute licence sur de tels droits pourrait être ou n'être pas disponible ; pas plus qu'elle ne prétend avoir accompli aucun effort pour identifier de tels droits. Les informations sur les procédures de l'ISOC au sujet des droits dans les documents de l'ISOC figurent dans les BCP 78 et BCP 79.

Des copies des dépôts d'IPR faites au secrétariat de l'IETF et toutes assurances de disponibilité de licences, ou le résultat de tentatives faites pour obtenir une licence ou permission générale d'utilisation de tels droits de propriété par ceux qui mettent en œuvre ou utilisent la présente spécification peuvent être obtenues sur répertoire en ligne des IPR de l'IETF à <http://www.ietf.org/ipr>.

L'IETF invite toute partie intéressée à porter son attention sur tous copyrights, licences ou applications de licence, ou autres droits de propriété qui pourraient couvrir les technologies qui peuvent être nécessaires pour mettre en œuvre la présente norme. Prière d'adresser les informations à l'IETF à ietf-ipr@ietf.org.

9. Remerciements

Le modèle d'informations de cœur de politique dans le présent document s'appuie étroitement sur le travail du groupe Accords de niveau de service du DMTF, de sorte que des remerciements sont dus aux membres de ce groupe de travail. Plusieurs des classes de politique de ce modèle sont apparues pour la première fois dans les premiers projets sur la politique IPsec et la politique de qualité de service. Les auteurs de ces projets étaient Partha Bhattacharya, Rob Adams, William Dixon, Roy Pereira, Raju Rajan, Jean-Christophe Martin, Sanjay Kamat, Michael See, Rajiv Chaudhury, Dinesh Verma, George Powers, et Raj Yavatkar. D'autres éléments du modèle trouvent leur origine dans le travail de Yoram Snir, Yoram Ramberg, et Ron Cohen. De plus, nous tenons à remercier Harald Alvestrand pour sa relecture attentive de ce document et ses nombreuses suggestions utiles, et Luis Sanchez et Russ Mundy pour leur aide sur les considérations pour la sécurité dans ce document.

10. Considérations pour la sécurité

Le modèle d'informations de cœur de politique (PCIM) présenté dans ce document fournit un modèle en mode objet pour décrire les informations de politique. Il fournit un cadre de base pour décrire la structure des informations de politique, dans une forme indépendante de tout répertoire ou protocole d'accès spécifique, à utiliser par un système d'exploitation. PCIM n'est pas destiné à représenter un concept ou mise en œuvre de système particulier, ni ne définit un protocole, et à ce titre il n'a pas d'exigence de sécurité spécifique.

Cependant, il devrait aussi être noté que certains documents dérivés, qui se fondent sur PCIM, auront besoin de comporter des considérations de sécurité plus spécifiques. Afin de communiquer la nature de ce qui sera attendu de ces documents dérivés, il est nécessaire de récapituler les raisons pour lesquelles PCIM, tel que défini dans le présent document, ne peut ni être mis en œuvre, ni être considéré comme représentatif, d'aucun système réel, ainsi que la nature des extensions et transpositions dérivées attendues.

Il y a trois raisons indépendantes pour lesquelles PCIM, tel que défini ici, n'est ni susceptible d'être mis en œuvre, ni représentatif d'aucun système réel :

1. Ses classes sont indépendantes de tout répertoire spécifique qui utilise un protocole d'accès spécifique. Donc, ses classes ne sont pas conçues pour être directement mises en œuvre. PCIM devrait plutôt être vu comme un schéma qui donne des indications sur la façon dont les informations devraient être représentées, indépendamment de toutes contraintes spécifiques de mise en œuvre de modèle.
2. Ses classes ont été conçues pour être indépendantes de tout domaine de politique spécifique. Par exemple, DiffServ et IPsec représentent deux domaines de politique différents. Chaque document qui étend PCIM à un de ces domaines va déduire des sous classes des classes et relations définies dans PCIM, afin de représenter les extensions d'un modèle générique pour couvrir les domaines techniques spécifiques.
3. C'est un modèle d'informations, qui doit être transposé en un modèle spécifique de données (schéma CIM natif, schéma LDAP, MIB, etc.) avant qu'il puisse être mis en œuvre. Les documents dérivés vont transposer le modèle d'informations étendu noté au point 2 ci-dessus en des types spécifiques de mise en œuvre de modèle de données.

Même si des exigences de sécurité spécifiques ne sont pas appropriées pour PCIM, des exigences de sécurité spécifiques DOIVENT être définies pour chaque application opérationnelle de PCIM dans la réalité. Tout comme il y a aura une large gamme de systèmes de fonctionnement réels qui utilisent PCIM, il y aura aussi une large gamme d'exigences de sécurité pour ces systèmes. Certains systèmes opérationnel réels qui sont déployés avec PCIM peuvent avoir de grandes exigences de sécurité qui impactent presque toutes les classes et sous classes utilisées par de tels systèmes, alors que les exigences de sécurité d'autres systèmes pourraient avoir très peu d'impact.

Les documents dérivés dont on parle plus haut vont créer le contexte d'application des exigences de sécurité opérationnelles réelles au niveau système à l'égard des divers modèles qui découlent de PCIM.

Par exemple, dans certains scénarios réels, les valeurs associées à certaines propriétés, au sein de certaines instances de classes, peuvent représenter des informations associées à des ressources rares, et/ou coûteuses (et donc précieuses). Il se peut que ces valeurs ne doivent pas être divulguées, ou manipulées par des personnes non autorisées. Tant que le modèle dérivé reste un modèle d'informations (par opposition à un modèle de données) il n'est pas possible de discuter des outils et mécanismes spécifiques du modèle de données qui sont disponibles pour réaliser l'authentification et l'autorisation implicite dans une exigence qui restreint l'accès en lecture et/ou lecture/écriture à ces valeurs. Donc, ces mécanismes vont devoir être discutés dans chacun des modèles de données auxquels les modèles d'informations dérivés sont transposés. Si il y a des exigences de sécurité générales qui peuvent être identifiées et peuvent être appliquées sur plusieurs types de

modèles de données, il serait approprié de les discuter au niveau du modèle d'informations, plutôt qu'à celui du modèle de données. Dans tous les cas, toute exigence de sécurité identifiée qui n'est pas traitée dans le document de modèle d'informations DOIT être traitée dans les documents dérivés du modèle de données.

On peut illustrer ces points en étendant l'exemple de la Section 2. Un système réel qui fournit la qualité de service Gold à John aurait vraisemblablement besoin de fournir au moins les capacités et mécanismes en rapport avec la sécurité suivants (voir dans [12] la définitions des termes en rapport avec la sécurité) :

- o L'intégrité des données pour les informations (par exemple, les valeurs des propriétés et les instances de relations) qui spécifient que John obtient le service de QS Gold, à partir du ou des points où les informations sont entrées dans le système jusqu'aux points où les composants de réseau fournissent en fait de service.
- o Les méthodes d'authentification et d'autorisation pour assurer que seuls les administrateurs système (et non John ou d'autres ingénieurs) peuvent administrer à distance les composants du système.
- o Une méthode d'authentification pour assurer que John reçoit le service Gold, et que les autres membres du groupe des ingénieurs reçoivent le service Bronze.

C'est un des ensembles possibles d'exigences associées à un exemple de système réel qui délivre le service Gold, et l'endroit approprié pour documenter cela serait une combinaison du modèle d'informations et des modèles de données dérivés pour une politique de QS. Chacun des modèles de données aurait aussi besoin d'expliquer comment les exigences sont satisfaites, en utilisant les mécanismes normalement disponibles pour un tel modèle de données, étant données la technologie ou les ensembles de technologies particulières qu'ils pourraient employer.

11. Références

- [1] Distributed Management Task Force, Inc., "DMTF Technologies: CIM Standards << CIM Schema: Version 2.4", disponible à partir de la page de la Toile de DMTF : http://www.dmtf.org/spec/cim_schema_v24.html .
- [2] Distributed Management Task Force, Inc., "Common Information Model (CIM) Specification, version 2.2", juin 1999. Disponible à : <http://www.dmtf.org/spec/cims.html> .
- [3] [RFC2119] S. Bradner, "[Mots clés à utiliser](#) dans les RFC pour indiquer les niveaux d'exigence", BCP 14, mars 1997.
- [4] [RFC2028] R. Hovey, S. Bradner, "Les organisations impliquées dans le processus de normalisation de l'IETF", octobre 1996. (MàJ par [RFC3668](#), [RFC3979](#)) ([BCP0011](#))
- [5] J. Strassner et S. Judd, "Directory-Enabled Networks", version 3.0c5 (août 1998). Un fichier PDF est disponible à <http://www.murchiso.com/den/#denspec>
- [6] J. Strassner, "Présentation de l'architecture de politique", 42^{ème} réunion de l'IETF, Chicago, Illinois, octobre 1998. Le compte-rendu de ce BOF est disponible à : <http://www.ietf.org/proceedings/98aug/index.html> .
- [7] [RFC2279] F. Yergeau, "UTF-8, un format de transformation de la norme ISO 10646", janvier 1998. (*Obsolète, voir [RFC3629](#)*) (*D.S.*)
- [8] [RFC2591] D. Levi, J. Schoenwaelder, "Définitions des objets gérés pour les opérations de gestion de planification", mai 1999. (*Obsolète, voir [RFC3231](#)*) (*P.S.*)
- [9] [RFC2591] D. Levi, J. Schoenwaelder, "Définitions des objets gérés pour les opérations de gestion de planification", mai 1999. (*Obsolète, voir [RFC3231](#)*) (*P.S.*)
- [10] [RFC2445] F. Dawson et D. Stenerson, "Spécification centrale des [objets de calendrier et de programmation](#) de l'Internet (iCalendar)", novembre 1998. (*P.S.*)
- [11] [RFC3703] J. Strassner et autres, "Schéma de cœur de politique du protocole léger d'accès à un répertoire (LDAP)", février 2004. (MàJ par [RFC4104](#)) (*P.S.*)
- [12] [RFC2828] R. Shirey, "Glossaire de la sécurité sur l'Internet", FYI 36, mai 2000. (*Obsolète, voir [RFC4949](#)*)

Note : La spécification du schéma CIM 2.4 est définie par l'ensemble de fichiers MOF disponibles à l'URL suivant : http://www.dmtf.org/spec/CIM_Schema24/CIM_Schema24.zip

12. Adresse des auteurs

Ed Elleson
LongBoard, Inc.
2505 Meridian Pkwy, #100
Durham, NC 27713
tél : +1 919-361-3230
fax: +1 919-361-3299
mél : elleson@lboard.com

Bob Moore
IBM Corporation, BRQA/502
4205 S. Miami Blvd.
Research Triangle Pk, NC 27709
tél : +1 919-254-4436
fax: +1 919-254-6243
mél : remoore@us.ibm.com

John Strassner
Cisco Systems, Bldg 15
170 West Tasman Drive
San Jose, CA 95134
tél : +1 408-527-1069
fax: +1 408-527-6351
mél : johns@cisco.com

Andrea Westerinen
Cisco Systems
170 West Tasman Drive
San Jose, CA 95134
tél : +1 408-853-8294
fax: +1 408-527-6351
mél : andreaw@cisco.com

13 Appendice A Identification de classe dans une mise en œuvre CIM native

Bien que la propriété `CommonName` soit présente dans la superclasse abstraite `Policy`, et soit donc disponible dans toutes ses sous classes instanciables, CIM n'utilise pas cette propriété pour nommer les instances. Les paragraphes qui suivent expliquent comment sont traitées les désignations dans une mise en œuvre native de CIM pour chacune des classes instanciables dans le modèle d'informations de cœur de politique.

Deux choses devraient être notées concernant les désignations de CIM :

- o Lorsque une association CIM est spécifiée comme "faible", c'est une déclaration sur les portées de désignation : une instance de la classe à l'extrémité faible de l'association est nommée au sein de la portée d'une instance de la classe à l'autre extrémité de l'association. Cela est accompli par la propagation des clés de l'instance de la classe de portée à l'instance de la classe faible. Donc, la classe faible a, via la propagation des clés, toutes les clés de la classe de portée, et elle a aussi une ou plusieurs clés supplémentaires pour distinguer les instances de la classe faible, dans le contexte de la classe de portée.
- o Tous les noms de classe dans CIM sont limités aux caractères alphabétiques et numériques plus le souligné, avec la restriction que le premier caractère ne peut pas être numérique. Se reporter à l'Appendice F "Usage d'Unicode" dans la référence [2] pour la spécification exacte de la façon dont les noms de classe CIM sont codés dans les chaînes CIM.

13.1 Instances de désignation de `PolicyGroup` et `PolicyRule`

Un groupe de politique existe toujours dans le contexte d'un système. Dans le modèle d'informations de cœur de politique, ceci est capturé par l'agrégation faible `PolicyGroupInSystem` entre un `PolicyGroup` et `System`. Noter que `System` sert de classe de base pour décrire les appareils du réseau et les domaines administratifs.

Une règle de politique existe aussi dans le contexte d'un système. Dans le modèle d'informations de cœur de politique, ceci est capturé par l'association faible `PolicyRuleInSystem` entre une `PolicyRule` et un `System`.

Les paragraphes qui suivent définissent les clés CIM pour `PolicyGroup` et `PolicyRule`.

13.1.1 Clés CIM de `PolicyGroup`

Les clés CIM de la classe `PolicyGroup` sont :

- o `SystemCreationClassName` (une clé CIM_System, propagée à cause de l'association faible, `PolicyGroupInSystem`)
- o `SystemName` (clé CIM_System, propagée à cause de l'association faible, `PolicyGroupInSystem`)
- o `CreationClassName`
- o `PolicyGroupName`

Elles sont définie dans la référence [1] comme suit :

NOM : `SystemCreationClassName`
DESCRIPTION : `SystemCreationClassName` représente le nom de classe de l'objet CIM `System` qui fournit la portée de désignation pour l'instance de `PolicyGroup`.
SYNTAXE : chaîne [MaxLen 256] (*MaxLen = Longueur maximale*)
QUALIFIEUR : clé

NOM : `SystemName`
DESCRIPTION : `SystemName` représente le nom individuel de l'objet `System` particulier, fournissant la portée de désignation pour l'instance de `PolicyGroup`.

SYNTAXE : chaîne [MaxLen 256]
 QUALIFIEUR : clé

NOM : CreationClassName
 DESCRIPTION : Cette propriété est réglée à "CIM_PolicyGroup", si l'objet PolicyGroup est directement instancié. Ou, si il est égal au nom de classe de la sous classe PolicyGroup qui est instanciée.

SYNTAXE : chaîne [MaxLen 256]
 QUALIFIEUR : clé

NOM : PolicyGroupName
 DESCRIPTION : Nom identifiant de ce groupe de politique.
 SYNTAXE : chaîne [MaxLen 256]
 QUALIFIEUR : clé

13.1.2 Clés CIM de PolicyRule

Les clés CIM de la classe PolicyRule sont :

- o SystemCreationClassName (clé CIM_System, propagée à cause de l'association faible PolicyRuleInSystem)
- o SystemName (clé CIM_System, propagée à cause de l'association faible PolicyRuleInSystem)
- o CreationClassName
- o PolicyRuleName

SystemCreationClassName et SystemName fonctionnent de la même façon que défini pour la classe PolicyGroup. Voir les détails au paragraphe 13.1.1.

Les deux autres propriétés sont définies dans la référence [1] comme suit :

NOM : CreationClassName
 DESCRIPTION : Cette propriété est réglée à "CIM_PolicyRule", si l'objet PolicyRule est instancié directement. Ou, il est égal au nom de classe de la sous classe PolicyRule qui est instanciée.
 SYNTAXE : chaîne [MaxLen 256]
 QUALIFIEUR : clé

NOM : PolicyRuleName
 DESCRIPTION : Nom qui identifie cette règle de politique.
 SYNTAXE : chaîne [MaxLen 256]
 QUALIFIEUR : clé

13.2 Instances de dénomination de PolicyCondition et ses sous classes

Les clés CIM de la classe PolicyCondition sont :

- o SystemCreationClassName
- o SystemName
- o PolicyRuleCreationClassName
- o PolicyRuleName
- o CreationClassName
- o PolicyConditionName

Noter qu'aucune des clés n'est définie comme propagée, bien qu'elles paraissent convenir à cette convention. La raison de cette différence est parce que (comme indiqué aux paragraphes 5.1 et 6.4) la classe PolicyCondition est utilisée pour représenter des conditions aussi bien réutilisables que spécifiques d'une règle. Ceci, à son tour, affecte quelles associations sont valides pour une instance de PolicyCondition, et comment cette instance est nommée.

Dans un monde idéal, une instance de la classe PolicyCondition serait délimitée soit par son PolicyRepository (pour une condition réutilisable) soit par sa PolicyRule (pour une condition spécifique d'une règle). Cependant, CIM a comme restriction qu'une certaine classe peut seulement être "faible" pour une autre classe (c'est-à-dire, définie par une association faible).

Pour fonctionner dans les restrictions de désignation CIM, il est nécessaire de "simuler" des associations faibles entre PolicyCondition et PolicyRule, et entre PolicyCondition et PolicyRepository, d'après une technique qu'on appelle propagation de clé manuelle. Strictement parlant, la propagation de clé manuelle n'est pas une propagation de clé du tout.

Mais elle a le même effet que la propagation de clés (la vraie), de sorte que le nom convient.

La Figure 9 illustre comment fonctionne la propagation manuelle dans le cas de PolicyCondition. (Noter que seules les propriétés de clés sont montrées pour chaque classe.) Sur la figure, la ligne composée de "I" indique l'héritage de classe, celle composée de "P" indique la propagation de clé (la vraie) via l'agrégation faible PolicyRuleInSystem, et celles composées de "M" indiquent la propagation de clé manuelle.

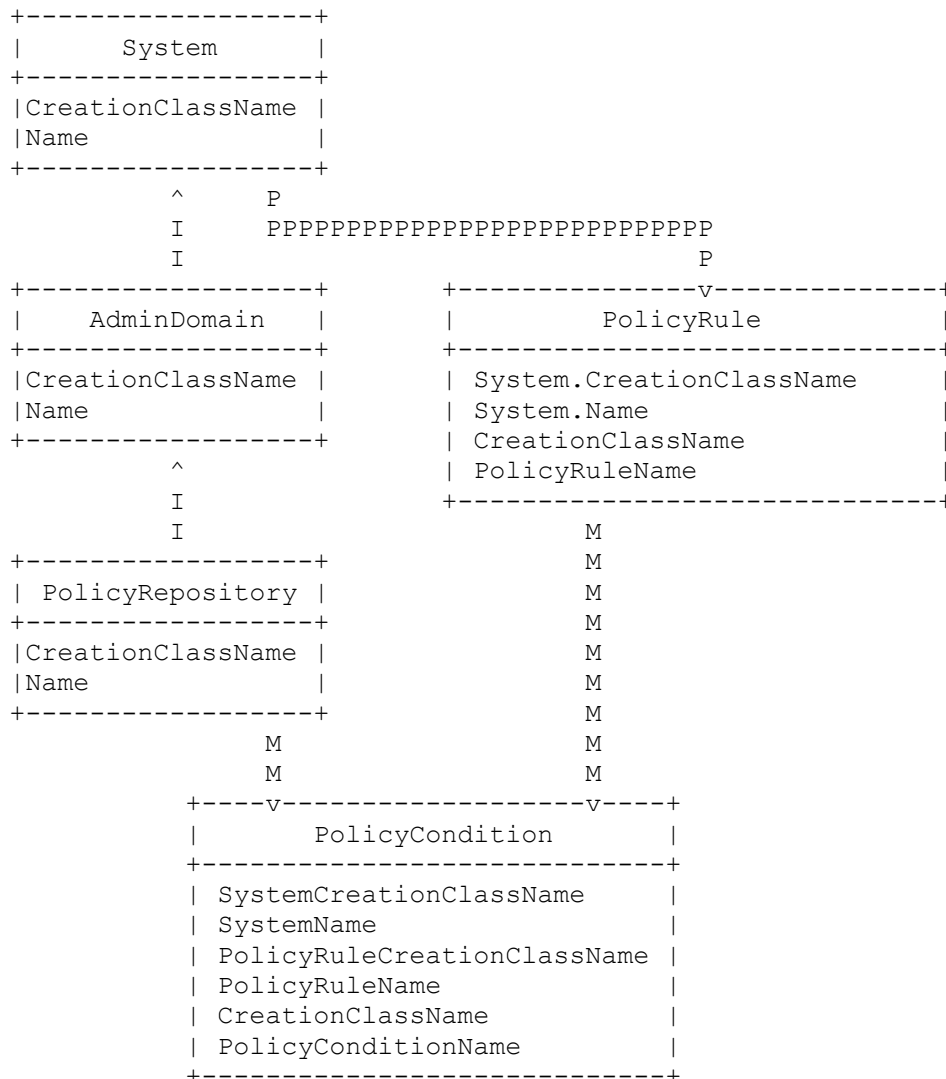


Figure 9 : Propagation manuelle de clé pour la désignation de PolicyCondition

Quand on regarde la Figure 9, on voit que deux propriétés de clé, CreationClassName et Name, sont définies dans la classe System, et héritées par ses sous classes AdminDomain et PolicyRepository. Comme PolicyRule est faible pour System, ces deux clés lui sont propagées ; il a aussi ses propres clés CreationClassName et PolicyRuleName.

Une approche similaire, bien que non automatique, est utilisée dans la "propagation de clé manuelle". Voici l'approche pour les PolicyCondition spécifiques d'une règle et réutilisables :

- o La propagation manuelle de clés de PolicyRule à PolicyCondition implique de copier les valeurs des quatre propriétés clés de PolicyRule en quatre propriétés clés désignées de façon similaire dans PolicyCondition. Du point de vue du langage de la spécification CIM, la propriété SystemName dans PolicyCondition est une propriété clé complètement nouvelle. Cependant, la relation à la propriété Name dans System est définie dans la description de SystemName.
- o La propagation manuelle de clés de PolicyRepository à PolicyCondition fonctionne exactement de la même façon pour les deux premières propriétés de clé. Cependant, comme PolicyRepository n'inclut pas de propriété PolicyRule, PolicyRuleCreationClassName et PolicyRuleName n'ont pas de valeurs. Une valeur spéciale, "No Rule", est allouée à ces deux propriétés dans ce cas, indiquant que cette instance de PolicyCondition n'est nommée dans la portée d'aucune règle de politique particulière.

Le paragraphe qui suit définit les clé CIM spécifiques pour PolicyCondition.

13.2.1 Clés CIM de PolicyCondition

Les propriétés clés de PolicyCondition sont définies dans la référence [1] comme suit :

NOM :	SystemCreationClassName
DESCRIPTION :	SystemCreationClassName représente le nom de classe de l'objet CIM System qui donne la portée de dénomination pour l'instance de PolicyCondition. Pour une condition de politique spécifique d'une règle, c'est le type de système (par exemple, le nom de la classe qui a créé cette instance) dans le contexte duquel la règle de politique est définie. Pour une condition de politique réutilisable, c'est réglé à "CIM_PolicyRepository", si l'objet PolicyRepository est instancié directement. Ou, il est égal au nom de classe de la sous-classe de PolicyRepository qui est instanciée.
SYNTAXE :	chaîne [MaxLen 256]
QUALIFIEUR :	clé
NOM :	SystemName
DESCRIPTION :	Nom de l'objet System dans la portée duquel cette condition de politique est définie. Cette propriété achève l'identification de l'objet System. Pour une condition de politique spécifique d'une règle, c'est le nom de l'instance du système dans le contexte duquel la règle de politique est définie. Pour une condition de politique réutilisable, c'est le nom de l'instance de PolicyRepository qui contient la condition de politique.
SYNTAXE :	chaîne [MaxLen 256]
QUALIFIEUR :	clé
NOM :	PolicyRuleCreationClassName
DESCRIPTION :	Pour une condition de politique spécifique d'une règle, cette propriété identifie le nom de classe de l'instance de règle de politique, dans la portée de laquelle cette instance de PolicyCondition existe. Pour une condition de politique réutilisable, cette propriété est réglée à une valeur spéciale, "No Rule", qui indique que cette instance de PolicyCondition n'est pas unique dans une règle de politique.
SYNTAXE :	chaîne [MaxLen 256]
QUALIFIEUR :	clé
NOM :	PolicyRuleName
DESCRIPTION :	Pour une condition de politique spécifique d'une règle, PolicyRuleName achève l'identification de l'objet PolicyRule auquel cette condition est associée. Pour une condition de politique réutilisable, une valeur spéciale, "No Rule", est utilisée pour indiquer que cette condition est réutilisable.
SYNTAXE :	chaîne [MaxLen 256]
QUALIFIEUR :	clé
NOM :	CreationClassName
DESCRIPTION :	Nom de classe de la sous classe PolicyCondition qui est instanciée.
SYNTAXE :	chaîne [MaxLen 256]
QUALIFIEUR :	clé
NOM :	PolicyConditionName
DESCRIPTION :	Nom identifiant cette condition de politique.
SYNTAXE :	chaîne [MaxLen 256]
QUALIFIEUR :	clé

13.3 Instances de désignation de PolicyAction et ses sous classes

Du point de vue des désignations, la classe PolicyAction et ses sous classes fonctionnent exactement comme la classe PolicyCondition et ses sous classes. Voir les détails aux paragraphes 13.2 et 13.2.1.

Spécifiquement, les clés CIM de PolicyAction sont :

- o SystemCreationClassName
- o SystemName
- o PolicyRuleCreationClassName
- o PolicyRuleName
- o CreationClassName
- o PolicyActionName

Elles sont définies comme suit dans la référence [1] :

NOM : SystemCreationClassName
 DESCRIPTION : SystemCreationClassName représente le nom de classe de l'objet CIM qui donne la portée de dénomination pour l'instance de PolicyAction. Pour une action de politique spécifique d'une règle, c'est le type de système (par exemple, le nom de la classe qui a créé cette instance) dans le contexte duquel la règle de politique est définie. Pour une action de politique réutilisable, c'est l'ensemble de "CIM_PolicyRepository", si l'objet PolicyRepository est instancié directement. Ou, il est égal au nom de classe de la sous classe de PolicyRepository qui est instanciée.
 SYNTAXE : chaîne [MaxLen 256]
 QUALIFIEUR : clé

NOM : SystemName
 DESCRIPTION : Nom de l'objet System dans la portée duquel cette action de politique est définie. Cette propriété achève l'identification de l'objet System. Pour une action de politique spécifique d'une règle, c'est le nom de l'instance du système dans le contexte duquel cette règle de politique est définie. Pour une action de politique réutilisable, c'est le nom de l'instance de PolicyRepository qui détient l'action de politique.
 SYNTAXE : chaîne [MaxLen 256]
 QUALIFIEUR : clé

NOM : PolicyRuleCreationClassName
 DESCRIPTION : Pour une action de politique spécifique d'une règle, cette propriété identifie le nom de classe de l'instance de règle de politique dans la portée de laquelle cette instance de PolicyAction existe. Pour une action de politique réutilisable, cette propriété est réglée à une valeur spéciale, "No Rule", qui indique que cette instance de PolicyAction n'est pas unique à une règle de politique.
 SYNTAXE : chaîne [MaxLen 256]
 QUALIFIEUR : clé

NOM : PolicyRuleName
 DESCRIPTION : Pour une action de politique spécifique d'une règle, PolicyRuleName achève l'identification de l'objet PolicyRule auquel cette action est associée. Pour une action de politique réutilisable, une valeur spéciale, "No Rule", est utilisée pour indiquer que cette action est réutilisable.
 SYNTAXE : chaîne [MaxLen 256]
 QUALIFIEUR : clé

NOM : CreationClassName
 DESCRIPTION : Nom de classe de la sous classe de PolicyAction qui est instanciée.
 SYNTAXE : chaîne [MaxLen 256]
 QUALIFIEUR : clé

NOM : PolicyActionName
 DESCRIPTION : Nom qui identifie cette action de politique.
 SYNTAXE : chaîne [MaxLen 256]
 QUALIFIEUR : clé

13.4 Instances de désignation de PolicyRepository

Une instance de PolicyRepository est désignée par les deux propriétés de clé CreationClassName et Name qu'elle hérite de sa superclasse AdminDomain. Ces propriétés sont en fait définies dans la super classe de AdminDomain, System, et ensuite héritées par AdminDomain.

Pour les instances de PolicyRepository elles-mêmes, la valeur de CreationClassName doit être "CIM_PolicyRepository". (On rappelle que pour la lisibilité, le préfixe "CIM_" a été omis de tous les noms de classes dans le présent document). Si une sous classe de PolicyRepository (peut-être QosPolicyRepository) est définie et instanciée, le nom de classe "CIM_QosPolicyRepository" est alors utilisé dans CreationClassName.

La propriété Name achève simplement l'identification de l'instance de PolicyRepository.

13.5 Rôle de la propriété `CreationClassName` pour la désignation

Pour donner un peu de souplesse à la désignation des instances, CIM utilise une propriété appelée `CreationClassName`. L'idée de `CreationClassName` est de donner une autre dimension qui puisse être utilisée pour éviter les collisions de noms, dans le cas spécifique d'instances qui appartiennent à deux sous classes différentes d'une superclasse commune. Un exemple va illustrer comment fonctionne `CreationClassName`.

Supposons que nous ayons des instances de deux différentes sous classes de `PolicyCondition`, `FrameRelayPolicyCondition` et `BgpPolicyCondition`, et que ces instances s'appliquent au même contexte. Si nous avons seulement la seule propriété clé `PolicyConditionName` disponible pour distinguer les deux instances, une collision résulterait alors de la désignation des deux instances avec la valeur clé `PCName = "PC-1"`. Donc, les administrateurs de politique de disciplines largement différentes auraient alors à coordonner leur désignation des `PolicyCondition` pour ce contexte.

Avec `CreationClassName`, les collisions de ce type peuvent être éliminées, sans exiger de coordination entre les administrateurs de politique. Les deux instances peuvent être distinguées en donnant à leur `CreationClassName` des valeurs différentes. Une instance est maintenant identifiée avec les deux clés

```
CreationClassName = "FrameRelayPolicyCondition" + PCName = "PC-1",
```

tandis que l'autre est identifiée par

```
CreationClassName = "BgpPolicyCondition" + PCName = "PC-1".
```

Chacune des classes instanciables dans le cœur de modèle comporte la propriété `CreationClassName` comme clé en plus de sa propre clé spécifique de classe.

13.6 Références d'objet

Aujourd'hui, toutes les associations CIM impliquent deux références d'objet. CIM décompose une référence d'objet en deux parties : une partie d'ordre supérieur qui identifie un gestionnaire d'objet et un espace de noms, et un chemin de modèle qui identifie une instance d'objet au sein d'un espace de noms. Le chemin de modèle, à son tour, peut être décomposé en un identifiant de classe d'objet et un ensemble de valeurs de clés nécessaires pour identifier une instance de cette classe.

Parce que l'identifiant de classe d'objet fait partie du chemin de modèle, une référence d'objet CIM est fortement typée. La référence d'objet `GroupComponent` dans l'association `PolicyGroupInPolicyGroup`, par exemple, peut seulement pointer sur une instance de `PolicyGroup`, ou sur une instance d'une sous classe de `PolicyGroup`. À la différence de LDAP où un pointeur DN est complètement non typé : il identifie (par DN) une entrée, mais ne fait aucune restriction sur la ou les classes d'objet de cette entrée.

Une importante différence entre les définitions de propriétés CIM et les définitions de type d'attribut LDAP a été identifiée précédemment à la Section 6 : alors qu'une définition de type d'attribut LDAP a une portée globale, une définition de propriété CIM ne s'applique qu'à la classe dans laquelle elle est définie. Donc, les propriétés qui ont le même nom dans deux classes différentes sont libres d'avoir des types de données différents. CIM tire parti de cette souplesse en permettant que le type de données d'une référence d'objet soit outrepassée dans une sous classe de l'association de classes dans laquelle elle était initialement définie.

Par exemple, la référence d'objet `GroupComponent` est définie dans la classe d'agrégation abstraite `PolicyComponent` comme étant une référence à une instance de la classe `Policy`. Ce type de données pour `GroupComponent` est alors outrepassé dans les sous classes de `PolicyComponent`. Dans `PolicyGroupInPolicyGroup`, par exemple, `GroupComponent` devient une référence à une instance de `PolicyGroup`. Mais dans `PolicyConditionInPolicyRule`, il devient une référence à une instance de `PolicyRule`. Bien sûr, il n'y a pas une liberté totale pour outrepasser les références d'objet. Afin de rester cohérent avec sa super classe abstraite, une sous classe de `PolicyComponent` peut seulement outrepasser `GroupComponent` comme référence à une sous classe de `Policy`. Une classe `Policy` est le contexte générique pour la référence `GroupComponent` dans `PolicyComponent`.

14. Appendice B : Format d'objet géré de cœur de politique

```
// =====  
// Titre : Spécification du format d'objet géré de cœur de politique version 2.4
```

```
// Nom de fichier : CIM_Policy24.MOF
// Version :      2.4
// Livraison :    0
// Description :  Les classes d'objet ci-dessous sont énumérées dans un ordre qui évite les références anticipées. Les objets
//               requis, définis par d'autres groupes de travail, sont omis.
// Date : 27/06/2000
// CIMCR516a – a enraciné les associations de modèle sous le composant de politique ou PolicyInSystem. Ont été
// corrigées PolicyCondition/PolicyActionInPolicyRepository des sous classes provenant de PolicyInSystem (similaire aux
// Groupes et rôles 'InSystem').
// =====
// Auteur : Groupe de travail d'accord de niveau de service (SLA, Service Level Agreement) DMTF
// =====
// Éléments
// =====
#pragma Locale ("en-US")
// =====
// Policy
// =====
[Résumé, Description ("Classe abstraite qui décrit les propriétés communes de toutes les sous classes en rapport avec les
règles de politique, telles que PolicyGroup, PolicyRule et PolicyCondition. Toutes les instances d'entités en rapport avec
les règles de politique seront créées à partir de sous classes de CIM_Policy. L'exception à cette assertion est
PolicyRepository qui est un type de CIM_System.") ]

Classe CIM_Policy : CIM_ManagedElement
{
[Description ("Nom convivial de cet objet en rapport avec la politique.") ]
chaîne CommonName ;
[Description ( "Dispositif de mots-clés pour caractériser/catégoriser des objets de politique. Les mots-clés sont d'un de
deux types :
o Mots-clés définis dans ce format d'objet géré (MOF, Managed Object Format) et les autres MOF, ou dans les livres
blancs de DMTF. Ces mots-clés fournissent un moyen indépendant du fabricant et de l'installation pour caractériser les
objets de politique.
o Mots-clés dépendants de l'installation pour caractériser les objets de politique. Les exemples incluent 'Ingénierie',
'Facturation', et 'Révision de décembre 2000'.
"Ce MOF définit les mots-clés suivants : 'INCONNU', 'CONFIGURATION', 'USAGE', 'SÉCURITÉ', 'SERVICE',
'MOTIVATION', 'INSTALLATION', et 'ÉVÉNEMENT'. Ces concepts sont auto explicatifs et sont exposés plus en détails
dans le livre blanc SLA/Policy. Un mot-clé supplémentaire est défini : 'POLITIQUE'. Le rôle de ce mot-clé est d'identifier
les instances en rapport avec la politique qui peuvent n'être autrement pas identifiables, dans certaines mises en œuvre. Le
mot-clé 'POLITIQUE' N'EST PAS mutuellement exclusif des autres mots-clés spécifiés ci-dessus.") ]

Chaîne PolicyKeywords [];
};

// =====
// PolicyComponent
// =====
[Association, Abstrait, Agrégation, Description ("CIM_PolicyComponent est une association générique utilisée pour établir
des relations 'partielles' entre les sous classes de "CIM_Policy. Par exemple, l'association PolicyConditionInPolicyRule
définit que des PolicyCondition font partie d'une PolicyRule.")
]
Classe CIM_PolicyComponent
{
[Agrégé, Clé, Description ( "Politique parente dans l'association.")
]
CIM_Policy REF GroupComponent;
[Clé, Description ( "Politique fille/partie dans l'association.")
]
CIM_Policy REF PartComponent;
};

// =====
// PolicyInSystem
// =====
```

[Association, Abstrait, Description ("CIM_PolicyInSystem est une association générique utilisée pour établir des relations de dépendance entre les politiques et les systèmes qui les détiennent. Ces systèmes peuvent être des ComputerSystem (*systèmes informatiques*) où les politiques 'fonctionnent', ou elles peuvent être des répertoires de politique où les politiques sont mémorisées. Ces relations sont similaires au concept de CIM_Services qui dépendent de CIM_System comme défini par le HostedService (*service hébergé*).

" La cardinalité est Max(1) pour la référence Antécédent/Système car les politiques peuvent seulement être hébergées dans au plus un contexte de système. Certaines sous classes de l'association vont repréciser cette définition pour rendre les politiques faibles pour les systèmes. D'autres sous classes de PolicyInSystem vont définir une relation facultative d'hébergement. Des exemples de chacune de ces relations sont les associations respectivement PolicyRuleInSystem et PolicyConditionInPolicyRepository."

```
]
Classe CIM_PolicyInSystem : CIM_Dependency
{
[Outrepassé ("Antécédent"), Max (1), Description ( "Le système d'hébergement." )
]
CIM_System REF Antécédent ;
[Outrepassé ("Dépendent"), Description ( "La politique hébergée." )
]
CIM_Policy REF Dépendent ;
};
```

```
// =====
// PolicyGroup
// =====
```

[Description ("Conteneur pour un ensemble de PolicyGroup en rapport les uns avec les autres ou ensemble de PolicyRule en rapport les unes avec les autres, mais pas les deux. Les PolicyGroup sont définis et désignés par rapport au CIM_System qui fournit leur contexte.")

```
]
Classe CIM_PolicyGroup : CIM_Policy
{
[Propagé ("CIM_System.CreationClassName"), Clé, MaxLen (256),
Description ("Le CreationClassName du système qui établit la portée." )
]
```

```
Chaîne SystemCreationClassName ;
[Propagé ("CIM_System.Name"), Clé, MaxLen (256),
Description ("Nom du système qui établit la portée." )
]
```

```
Chaîne SystemName ;
[Clé, MaxLen (256), Description ( "CreationClassName indique le nom de la classe ou de la sous classe utilisée dans la
création d'une instance. Lorsque utilisé avec les autres propriétés clés de cette classe, cette propriété permet d'identifier de
façon univoque toutes les instances de cette classe et ses sous classes." ) ]
```

```
Chaîne CreationClassName ;
[Clé, MaxLen (256), Description ( "Nom convivial de ce PolicyGroup." )
]
```

```
Chaîne PolicyGroupName ;
};
```

```
// =====
// PolicyGroupInPolicyGroup
// =====
```

[Association, Agrégation, Description ("Relation qui agrège un ou plusieurs PolicyGroup de niveau inférieur en un groupe de niveau supérieur. Un PolicyGroup peut agréger des PolicyRule ou d'autres PolicyGroup, mais pas les deux.")

```
]
Classe CIM_PolicyGroupInPolicyGroup : CIM_PolicyComponent
{
[Outrepassé ("GroupComponent"), Agrège, Description ( "PolicyGroup qui agrège d'autres groupes." )
]
```

```
CIM_PolicyGroup REF GroupComponent ;
[Outrepassé ("PartComponent"), Description ( "PolicyGroup agrégé par un autre groupe." )
]
```

```
CIM_PolicyGroup REF PartComponent;
};
```

```
// =====
// PolicyGroupInSystem
// =====
[Association, Description ( "Association qui lie un PolicyGroup au système dans la portée duquel le groupe est défini." )
]
Classe CIM_PolicyGroupInSystem : CIM_PolicyInSystem
{
[Outrepassé ("Antécédent"), Min(1), Max(1), Description ( "Système dans la portée duquel un PolicyGroup est défini." )
]
CIM_System REF Antécédent;
[Outrepassé ("Dépendant"), Faible, Description ( "PolicyGroup nommé dans la portée d'un système." )
]
CIM_PolicyGroup REF Dépendant ;
};

// =====
// PolicyRule
// =====
[Description ( " Classe centrale pour représenter la sémantique 'Si condition alors action' associée à une règle de politique.
Une condition de PolicyRule, au sens le plus général, est représentée soit par un ensemble OUIxé de conditions ETixées
(forme disjonctive normale (DNF, Disjunctive Normal Form)) ou un ensemble ETixé de conditions OUIxées (forme
conjonctive normale (CNF, Conjunctive Normal Form)). Les conditions individuelles peuvent soit être niées (NON C) soit
non niées (C). Les actions spécifiées par une PolicyRule ne sont à effectuer que si et seulement si la condition de
PolicyRule (qu'elle soit représentée en DNF ou en CNF) s'évalue à VRAI.
" "
Les conditions et actions associées à une PolicyRule sont modélisées, respectivement, avec des sous classes de
PolicyCondition et PolicyAction. Ces objets de condition et action sont liés aux instances de PolicyRule par les agrégations
PolicyConditionInPolicyRule et PolicyActionInPolicyRule.
" "
Une PolicyRule peut aussi être associée à une ou plusieurs périodes de politique, indiquant le programme selon lequel la
règle de politique est active et inactive. Dans ce cas, c'est l'agrégation PolicyRuleValidityPeriod qui fournit ce lien.
" "
La classe PolicyRule utilise la propriété ConditionListType, pour indiquer si les conditions pour la règle sont en DNF ou en
CNF. L'agrégation PolicyConditionInPolicyRule contient deux propriétés supplémentaires pour achever la représentation
de l'expression conditionnelle de la règle. La première de ces propriétés est un entier pour partager les PolicyCondition
référéncées en un ou plusieurs groupes, et la seconde est un booléen pour indiquer si une condition référéncée est niée. Un
exemple montre comment ConditionListType et ces deux propriétés supplémentaires fournissent une représentation
univoque d'un ensemble de PolicyCondition en DNF ou en CNF.
" "
Supposons que nous ayons une PolicyRule qui agrège cinq PolicyCondition C1 à C5, avec les valeurs suivantes dans les
propriétés des cinq associations PolicyConditionInPolicyRule :
C1 : GroupNumber = 1, ConditionNegated = FAUX
C2 : GroupNumber = 1, ConditionNegated = VRAI
C3 : GroupNumber = 1, ConditionNegated = FAUX
C4 : GroupNumber = 2, ConditionNegated = FAUX
C5 : GroupNumber = 2, ConditionNegated = FAUX
Si ConditionListType = DNF, la condition globale pour la PolicyRule est alors :
(C1 ET (NON C2) ET C3) OU (C4 ET C5)
" "
D'un autre côté, si ConditionListType = CNF, la condition globale pour la PolicyRule est alors :
(C1 OU (NON C2) OU C3) ET (C4 OU C5)
" "
Dans les deux cas, il y a une spécification non ambiguë de la condition globale qui est vérifiée pour déterminer si on
effectue les PolicyAction associées à la PolicyRule.)
]
Classe CIM_PolicyRule : CIM_Policy
{
[Propagée ("CIM_System.CreationClassName"),
Clé, MaxLen (256), Description ("Le CreationClassName du système qui définit la portée.")
]
Chaîne SystemCreationClassName ;
[Propagée ("CIM_System.Name"),
```

```

    Clé, MaxLen (256),
    Description ("Nom du système qui définit la portée.")
]
Chaîne SystemName ;
[Clé, MaxLen (256), Description ( "CreationClassName indique le nom de la classe ou de la sous classe utilisée dans la
création d'une instance. Lorsque utilisée avec les autres propriétés clés de cette classe, cette propriété permet que toutes les
instances de cette classe et de ses sous classes soient identifiées de façon univoque." )]
Chaîne CreationClassName ;
[Clé, MaxLen (256), Description ( "Nom convivial de cette PolicyRule." )]
]
Chaîne PolicyRuleName ;
[Description ( "Indique si cette PolicyRule est activée administrativement, désactivée administrativement, ou activée pour
un débogage. Lorsque la propriété a la valeur 3 ("enabledForDebug"), l'entité qui évalue les PolicyCondition a pour
instruction d'évaluer les conditions pour la règle, mais pas d'effectuer les actions si les PolicyCondition s'évaluent à VRAI.
Cela sert de véhicule de débogage lorsque on tente de déterminer quelles politiques vont s'exécuter dans un scénario
particulier, sans entreprendre aucune action pour changer d'état durant le débogage. La valeur par défaut est 1
("activé")."),
    ValueMap { "1", "2", "3" },
    Valeurs { "enabled", "disabled", "enabledForDebug" }
]
uint16 Enabled;
[Description ( "Indique si la liste des PolicyConditions associées à cette PolicyRule est en forme disjonctive normale
(DNF) ou en forme conjonctive normale (CNF). La valeur par défaut est 1 ("DNF")."),
    ValueMap { "1", "2" },
    Valeurs { "DNF", "CNF" }
]
uint16 ConditionListType;
[Description ( "Chaîne de forme libre qui peut être utilisée pour donner des lignes directrices sur la façon d'utiliser cette
PolicyRule." )]
]
Chaîne RuleUsage ;
[Description ( "Entier non négatif pour donner une priorité à cette PolicyRule par rapport aux autres règles. Une plus
grande valeur indique une plus forte priorité. La valeur par défaut est 0." )]
]
uint16 Priority ;
[Description ( "Fonion qui indique que l'évaluation des PolicyCondition et l'exécution des PolicyAction (si les conditions
s'évaluent à VRAI) sont exigées. L'évaluation d'une PolicyRule DOIT être tentée si la valeur de propriété Obligatoire est
VRAI. Si la propriété Obligatoire est FAUX, alors l'évaluation de la règle est 'au mieux' et PEUT être ignorée." )]
]
booléen Obligatoire ;
[Description ( "Cette propriété donne à un administrateur de politique un moyen de spécifier comment doit être interprété
l'ordre des PolicyAction associées à cette PolicyRule. Trois valeurs sont acceptées :
o obligatoire(1) : Faire les actions dans l'ordre indiqué, ou ne pas les faire du tout."
o recommandé(2) : Faire les actions dans l'ordre indiqué si on peut, mais si on ne peut pas les faire dans cet ordre, les
faire si on le peut dans un autre ordre."
o n'importe(3) : Les faire – on ne se soucie pas de l'ordre.
La valeur par défaut est 3 ("n'importe")."),
    ValueMap { "1", "2", "3" },
    Valeurs { "obligatoire", "recommandé", "n'importe" }
]
uint16 SequencedActions;
[Description ( "Cette propriété représente les rôles et combinaisons de rôles associés à une PolicyRule. Chaque valeur
représente un rôle ou une combinaison de rôles. Comme c'est une propriété multi valeurs, plus d'un rôle ou combinaison
peut être associé à une seule règle de politique. Chaque valeur est une chaîne de la forme :
<RoleName>[&&<RoleName>]*
où les noms de rôles individuels apparaissent en ordre alphabétique (selon la séquence de collation de l'UCS-2).
]
Chaîne PolicyRoles [] ;
};

// =====
// PolicyRuleInPolicyGroup
// =====

```

```

[Association, Agrégation, Description ( "Relation qui agrège une ou plusieurs PolicyRule dans un PolicyGroup. Un
PolicyGroup peut agréger soit des PolicyRule soit d'autres PolicyGroup, mais pas les deux.")
]
Classe CIM_PolicyRuleInPolicyGroup : CIM_PolicyComponent
{
[Outrepassse ("GroupComponent"), Agrégé, Description ( "PolicyGroup qui agrège une ou plusieurs PolicyRule.")
]
CIM_PolicyGroup REF GroupComponent ;
[Outrepassse ("PartComponent"), Description ( "PolicyRule agrégée par un PolicyGroup.")
]
CIM_PolicyRule REF PartComponent ;
};

// =====
// PolicyRuleInSystem
// =====
[Association, Description ( "Association qui relie une PolicyRule au système dans la portée duquel la règle est définie.")
]
Classe CIM_PolicyRuleInSystem : CIM_PolicyInSystem
{
[Outrepassse ("Antécédent"), Min(1), Max(1), Description ( "Système dans la portée duquel une PolicyRule est définie.")
]
CIM_System REF Antécédent ;
[Outrepassse ("Dépendant"), Faible, Description ( "PolicyRule nommée au sein de la portée d'un System.")
]
CIM_PolicyRule REF Dépendant ;
};

// =====
// PolicyRepository
// =====
[Description ( "Classe qui représente un conteneur défini administrativement pour des informations en rapport avec une
politique réutilisable. Cette classe n'introduit aucune propriété supplémentaire au delà de celles de sa super classe
AdminDomain. Elle participe cependant à un certain nombre d'associations uniques."

Une instance de cette classe utilise la valeur NameFormat "\"PolicyRepository\"", qui est définie dans la classe
AdminDomain.")
]
Classe CIM_PolicyRepository : CIM_AdminDomain
{
};

// =====
// PolicyRepositoryInPolicyRepository
// =====
[Association, Agrégation, Description ( "Relation qui agrège un ou plusieurs PolicyRepository de niveau inférieur dans
un répertoire de niveau supérieur.")
]
Classe CIM_PolicyRepositoryInPolicyRepository : CIM_SystemComponent
{
[Outrepassse ("GroupComponent"), Agrégé, Description ( "PolicyRepository qui agrège d'autres répertoires.")
]
CIM_PolicyRepository REF GroupComponent;
[Outrepassse ("PartComponent"), Description ( "PolicyRepository agrégé par un autre répertoire.")
]
CIM_PolicyRepository REF PartComponent ;
};

// =====
// PolicyCondition
// =====
[Abstrait, Description ( "Classe qui représente une condition de politique spécifique d'une règle ou réutilisable à évaluer en
conjonction avec une PolicyRule. Comme tous les détails du fonctionnement d'une PolicyCondition sont fournis dans les

```

sous classes de cet objet, cette classe est abstraite.")

]

Classe CIM_PolicyCondition : CIM_Policy

{

[Clé, MaxLen (256), Description ("Nom de la classe ou sous classe utilisée dans la création de l'objet System dans la portée duquel est définie cette PolicyCondition."

" "

"Cette propriété aide à identifier l'objet System dans la portée duquel existe cette instance de PolicyCondition. Pour une PolicyCondition spécifique d'une règle, c'est le système dans le contexte duquel est définie la PolicyRule. Pour une PolicyCondition réutilisable, c'est l'instance de PolicyRepository (qui est une sous classe de System) qui détient la condition."

" "

"Noter que cette propriété, et la propriété SystemName analogue, ne représentent pas les clés propagées d'une instance de la classe System. Ce sont plutôt les propriétés définies dans le contexte de cette classe, qui répètent les valeurs de l'instance du système auquel se rapporte cette PolicyCondition, soit directement via l'agrégation PolicyConditionInPolicyRepository, soit indirectement via l'agrégation PolicyConditionInPolicyRule."

]

Chaîne SystemCreationClassName ;

[Clé, MaxLen (256), Description ("Nom de l'objet System dans la portée duquel est définie cette PolicyCondition."

" "

"Cette propriété achève l'identification de l'objet System dans la portée duquel existe cette instance de PolicyCondition. Pour une PolicyCondition spécifique d'une règle, c'est le système dans le contexte duquel est définie la PolicyRule. Pour une PolicyCondition réutilisable, c'est l'instance de PolicyRepository (qui est une sous classe de System) qui détient la condition."

]

Chaîne SystemName ;

[Clé, MaxLen (256), Description ("Pour une PolicyCondition spécifique d'une règle, le CreationClassName de l'objet PolicyRule auquel cette condition est associée. Pour une PolicyCondition réutilisable, une valeur spéciale, 'NO RULE', devrait être utilisée pour indiquer que cette condition est réutilisable et non associée à une seule PolicyRule."

]

Chaîne PolicyRuleCreationClassName ;

[Clé, MaxLen (256), Description ("Pour une PolicyCondition spécifique d'une règle, le nom de l'objet PolicyRule auquel cette condition est associée. Pour une PolicyCondition réutilisable, une valeur spéciale, 'NO RULE', devrait être utilisée pour indiquer que cette condition est réutilisable et non associée à une seule PolicyRule.")

]

Chaîne PolicyRuleName ;

[Clé, MaxLen (256), Description ("CreationClassName indique le nom de la classe ou des sous classes utilisées dans la création d'une instance. Lorsque utilisée avec les autres propriétés clé de cette classe, cette propriété permet d'identifier de façon univoque toutes les instances de cette classe et ses sous classes.")

]

Chaîne CreationClassName ;

[Clé, MaxLen (256), Description ("Nom convivial de cette PolicyCondition."

]

Chaîne PolicyConditionName ;

};

// =====

// PolicyConditionInPolicyRule

// =====

[Association, Agrégation, Description ("Une PolicyRule agrège zéro, une ou plusieurs instances de la classe PolicyCondition, via l'association PolicyConditionInPolicyRule. Une règle qui agrège zéro condition n'est pas valide – elle peut, cependant, être en cours d'entrée dans un PolicyRepository ou en cours de définition pour un système. Noter qu'une PolicyRule ne devrait pas avoir d'effet tant qu'elle n'est pas valide."

" "

"Les conditions agrégées par une PolicyRule sont groupées dans deux niveaux de listes : soit un ensemble OUIxé d'ensembles ETixés de conditions (DNF, par défaut) ou un ensemble ETixé d'ensembles OUIxés de conditions (CNF). Les PolicyCondition individuelles dans ces listes peuvent être niées. La propriété ConditionListType spécifie lequel de ces deux schémas de groupage s'applique à une PolicyRule particulière."

" "

"Dans l'un et l'autre cas, les PolicyCondition sont utilisées pour déterminer si il faut effectuer les PolicyAction associées à la PolicyRule."

" "

"Une ou plusieurs PolicyTimePeriodCondition peuvent être parmi les conditions associées à une PolicyRule via

l'association PolicyConditionInPolicyRule. Dans ce cas, les périodes de temps sont simplement des conditions supplémentaires à évaluer avec toutes les autres qui sont spécifiées pour la règle. ")

```

]
Classe CIM_PolicyConditionInPolicyRule : CIM_PolicyComponent
{
[Outrepassé ("GroupComponent"), Agrège, Description ( "Cette propriété représente la PolicyRule qui contient une ou
plusieurs PolicyCondition.")
]
CIM_PolicyRule REF GroupComponent;
[Outrepassé ("PartComponent"), Description ( "Cette propriété détient le nom d'une PolicyCondition contenue par une ou
plusieurs PolicyRule.")
]
CIM_PolicyCondition REF PartComponent;
[Description ( "Entier non signé qui indique le groupe auquel appartient la PolicyCondition identifiée par la propriété
ContainedCondition. Cet entier segmente les conditions en ensembles ETixés (lorsque le ConditionListType est "\"DNF\"")
ou de façon similaire en ensembles OUIxés (lorsque le ConditionListType est "\"CNF\"") qui sont alors évalués.")
]
uint16 GroupNumber;
[Description ( "Indication de si la condition identifiée par la propriété ContainedCondition est niée. VRAI indique que la
PolicyCondition EST niée, FAUX indique qu'elle N'EST PAS niée.")
]
booléen ConditionNegated ;
};

```

```

// =====
// PolicyConditionInPolicyRepository
// =====
[Association, Description ( " Classe qui représente l'hébergement des PolicyCondition réutilisables par un
PolicyRepository. Une PolicyCondition réutilisable se rapporte toujours à un seul PolicyRepository, via cette agrégation."
" "
"Noter qu'une instance de PolicyCondition peut être réutilisable ou spécifique d'une règle. Lorsque la condition est
spécifique d'une règle, elle ne doit pas se rapporter à un PolicyRepository via l'agrégation
PolicyConditionInPolicyRepository."
]
Classe CIM_PolicyConditionInPolicyRepository : CIM_PolicyInSystem
{
[Outrepassé ("Antécédent"), Max(1), Description ( "Cette propriété identifie un PolicyRepository hébergeant une ou
plusieurs PolicyCondition. Une PolicyCondition réutilisable se rapporte toujours exactement à un PolicyRepository via
l'agrégation PolicyConditionInPolicyRepository. La cardinalité [0..1] pour cette propriété couvre les deux types de
PolicyCondition : 0 pour une PolicyCondition spécifique d'une règle, 1 pour une réutilisable.")
]
CIM_PolicyRepository REF Antécédent;
[Outrepassé ("Dépendent"), Description ( "Cette propriété détient le nom d'une PolicyCondition hébergée dans le
PolicyRepository. ")
]
CIM_PolicyCondition REF Dépendent;
};

```

```

// =====
// PolicyTimePeriodCondition
// =====
[Description ( " Cette classe donne un moyen pour représenter les périodes durant lesquelles une PolicyRule est valide,
c'est-à-dire, active. À tous les instants qui tombent en-dehors de ces périodes, la PolicyRule n'a pas d'effet. Une règle est
traitée comme valide à TOUT instant si elle ne spécifie pas une PolicyTimePeriodCondition."
" "
"Dans certains cas, un consommateur de politique peut avoir besoin d'effectuer certaines actions de réglage/nettoyage
lorsque une PolicyRule devient active/inactive. Par exemple, des sessions qui ont été établies alors qu'une règle était active
peuvent devoir être interrompues lorsque la règle devient inactive. Dans d'autres cas cependant, de telles sessions
pourraient être laissées actives. Dans ce cas, l'effet de la désactivation de la PolicyRule serait juste d'empêcher
l'établissement de nouvelles sessions."
" "
"Les comportements d'établissement/nettoyage pendant les transitions de période de validité ne sont pas traitées
actuellement par le modèle de politique, et doivent être spécifiés dans des documents donnant des lignes directrices ou via

```

des sous classes de CIM_PolicyRule, CIM_PolicyTimePeriodCondition ou d'autres sous classes concrètes de CIM_Policy. Si de tels comportements doivent être sous le contrôle de l'administrateur de politique, un mécanisme permettant le contrôle doit alors aussi être spécifié dans les sous classes."

" "

"PolicyTimePeriodCondition est défini comme une sous classe de PolicyCondition. C'est pour permettre l'inclusion de critères fondés sur le temps dans les définitions de condition ET/OU pour une PolicyRule."

" "

"Les instances de cette classe peuvent avoir jusqu'à cinq propriétés qui identifient les périodes à différents niveaux. Les valeurs de toutes les propriétés présentes dans une instance sont ETixées ensemble pour déterminer la ou les périodes de validité pour l'instance. Par exemple, une instance avec une gamme de validité globale du 1^{er} janvier au 31 décembre 2000, un gabarit mensuel qui choisit mars et avril, un gabarit de jour de la semaine qui choisit les vendredis, et une gamme d'heures de 0800 à 1600 seraient représentés en utilisant les périodes suivantes :

" vendredi 5 mars 2000, de 0800 à 1600 ; "
 " vendredi 12 mars 2000, de 0800 à 1600 ; "
 " vendredi, 19 mars 2000, de 0800 à 1600 ; "
 " vendredi 26 mars 2000, de 0800 à 1600 ; "
 " vendredi 2 avril 2000, de 0800 à 1600 ; "
 " vendredi 9 avril 2000, de 0800 à 1600 ; "
 " vendredi 16 avril 2000, de 0800 à 1600 ; "
 " vendredi 23 avril 2000, de 0800 à 1600 ; "
 " vendredi 30 avril 2000, de 0800 à 1600."
 " "

"Les propriétés qui ne sont pas présentes dans une instance de PolicyTimePeriodCondition sont implicitement traitées comme ayant leur valeur 'toujours activée'. Donc, dans l'exemple ci-dessus, le gabarit du jour dans le mois n'est pas présent, et la période de validité pour l'instance inclut implicitement un gabarit de jour du mois qui choisit tous les jours du mois. Si cette règle de la 'propriété manquante' est appliquée complètement, on voit qu'il y a une deuxième façon d'indiquer qu'une PolicyRule est toujours activée : lui associer une instance de PolicyTimePeriodCondition dont seules les propriétés avec des valeurs spécifiques sont ses propriétés clés.")

]

Classe CIM_PolicyTimePeriodCondition : CIM_PolicyCondition

{

[Description (" Cette propriété identifie une gamme globale de dates calendaires et d'heures sur lesquelles une PolicyRule est valide. Elle est formatée comme une chaîne représentant une date et heure de début, dans laquelle le caractère 'T' indique le début de la portion horaire, suivi par le caractère barre oblique '/', suivi par une chaîne similaire représentant une date et heure de fin. La première date indique le début de la gamme, et la seconde date en indique la fin. Donc, la seconde date et heure doit être postérieure à la première. Les dates et heures sont exprimées par des sous chaîne de la forme aaaammjjThhmmss. Par exemple :

20000101T080000/20000131T120000 définit du 1^{er} janvier 2000, 0800 au 31 janvier 2000, midi"

" "

"Il y a aussi deux cas particuliers dans lesquels une des chaînes date/heure est remplacée par une chaîne spéciale définie dans la RFC 2445."

- o Si la première date/heure est remplacée par la chaîne 'THISANDPRIOR', la propriété indique alors qu'une PolicyRule est valide [de maintenant] jusqu'à la date/heure qui apparaît après le caractère '/'.
- o Si la seconde date/heure est remplacée par la chaîne 'THISANDFUTURE', la propriété indique alors qu'une PolicyRule devient valide à la date/heure qui apparaît avant le '/', et reste valide à partir de ce point. "),

ModelCorrespondence {

"CIM_PolicyTimePeriodCondition.MonthOfYearMask",
 "CIM_PolicyTimePeriodCondition.DayOfMonthMask",
 "CIM_PolicyTimePeriodCondition.DayOfWeekMask",
 "CIM_PolicyTimePeriodCondition.TimeOfDayMask",
 "CIM_PolicyTimePeriodCondition.LocalOrUtcTime"}
]

Chaîne TimePeriod ;

[Chaîne d'octets, Description ("L'objet de cette propriété est de préciser la période de validité qui est définie par la propriété TimePeriod, en spécifiant explicitement dans quels mois la PolicyRule est valide. Ces propriétés fonctionnent ensemble, avec la TimePeriod utilisée pour spécifier la période globale dans laquelle la PolicyRule est valide, et le MonthOfYearMask utilisé pour retenir les mois durant lesquels la règle est valide."
 " "

" "

"Cette propriété est formatée comme une chaîne d'octets, structurée comme suit :"

- o un champ de 4 octets, qui indique la longueur de la chaîne d'octets entière ; ce champ est toujours réglé à 0x00000006 pour cette propriété ;"
- o un champ de 2 octets consistant en 12 bits identifiant les 12 mois de l'année, commençant par janvier et finissant par décembre, suivis par 4 bits qui sont toujours réglés à '0'. Pour chaque mois, la valeur '1' indique que la politique est

valide pour ce mois, et la valeur '0' indique qu'elle n'est pas valide.

" "

"La valeur 0x000000060830, par exemple, indique qu'une PolicyRule n'est valide que les mois de mai, novembre, et décembre.

" "

"Si une valeur n'est pas fournie pour cette propriété, la PolicyRule est alors traitée comme valide pour les douze mois, et n'est restreinte que par sa valeur de propriété TimePeriod et les autres propriétés de gabarit."),

ModelCorrespondence {

"CIM_PolicyTimePeriodCondition.TimePeriod",

"CIM_PolicyTimePeriodCondition.LocalOrUtcTime"}

]

uint8 MonthOfYearMask[];

[Chaîne d'octets, Description ("L'objet de cette propriété est de préciser la période de validité qui est définie par la propriété TimePeriod, en spécifiant explicitement les jours du mois où la PolicyRule est valide. Ces propriétés fonctionnent ensemble, avec la TimePeriod utilisée pour spécifier la période globale dans laquelle la PolicyRule est valide, et le DayOfMonthMask utilisé pour retenir les jours du mois pendant lesquels la règle est valide.

" "

"Cette propriété est formatée comme chaîne d'octets, structurée comme suit :

- o un champ de 4 octets qui indique la longueur de la chaîne d'octets entière ; ce champ est toujours réglé à 0x0000000C pour cette propriété ;
- o un champ de 8 octets consistant en 31 bits qui identifient les jours du mois en comptant depuis le début, suivis par 31 autres bits qui identifient les jours du mois en comptant depuis la fin, suivis par 2 bits qui sont toujours à '0'. Pour chaque jour, la valeur '1' indique que la politique est valide ce jour, et la valeur '0' indique qu'elle n'est pas valide."

" "

"La valeur 0x0000000C8000000100000000, par exemple, indique qu'une PolicyRule est valide le premier et le dernier jour du mois."

" "

"Pour les mois qui ont moins de 31 jours, les chiffres correspondant aux jours que n'a pas le mois (en comptant dans les deux directions) sont ignorés."

" "

"Si une valeur n'est pas fournie pour cette propriété, la PolicyRule est alors traitée comme valide pour tous les jours du mois, et n'est restreinte que par la valeur de sa propriété TimePeriod et les autres propriétés du gabarit."),

ModelCorrespondence {

"CIM_PolicyTimePeriodCondition.TimePeriod",

"CIM_PolicyTimePeriodCondition.LocalOrUtcTime"}

]

uint8 DayOfMonthMask[];

[Chaîne d'octets, Description ("L'objet de cette propriété est de préciser la période de validité qui est définie par la propriété TimePeriod, en spécifiant explicitement quels jours du mois la PolicyRule est valide. Ces propriétés fonctionnent ensemble, avec la TimePeriod utilisée pour spécifier la période globale dans laquelle la PolicyRule est valide, et le DayOfWeekMask utilisé pour retenir les jours de la semaine pendant lesquels la règle est valide."

" "

"Cette propriété est formatée comme chaîne d'octets, structurée comme suit :

- o champ de 4 octets, indiquant la longueur de la chaîne d'octets entière ; ce champ est toujours réglé à 0x00000005 pour cette propriété ;"
- o champ d'un octet consistant en 7 bits qui identifient les 7 jours de la semaine, commençant par dimanche et se terminant à samedi, suivis par un bit qui est toujours réglé à '0'. Pour chaque jour de la semaine, la valeur '1' indique que la politique est valide ce jour, et la valeur '0' indique qu'elle n'est pas valide."

" "

"La valeur 0x000000057C, par exemple, indique qu'une PolicyRule est valide du lundi au vendredi."

" "

"Si une valeur n'est pas fournie pour cette propriété, la PolicyRule est alors traitée comme valide pour tous les jours de la semaine et n'est restreinte que par la valeur de sa propriété TimePeriod et les autres propriétés du gabarit."),

ModelCorrespondence {

"CIM_PolicyTimePeriodCondition.TimePeriod",

"CIM_PolicyTimePeriodCondition.LocalOrUtcTime"}

]

uint8 DayOfWeekMask[] ;

[Description (" L'objet de cette propriété est de préciser la période de validité qui est définie par la propriété TimePeriod, en spécifiant explicitement une gamme d'heures de la journée pendant lesquelles la PolicyRule est valide. Ces propriétés fonctionnent ensemble, avec la TimePeriod utilisée pour spécifier la période globale pendant laquelle la PolicyRule est valide, et le TimeOfDayMask utilisé pour retenir la gamme de périodes d'un certain jour durant laquelle la règle est valide.

" "

"Cette propriété est formatée dans le style de la RFC 2445 : une chaîne horaire commençant par le caractère 'T', suivie par le caractère barre oblique '/', suivi par une seconde chaîne horaire. La première heure indique le début de la gamme, la seconde heure en indique la fin. Les heures sont exprimées par des sous chaînes de la forme 'Thhmmss'."

" "

"La seconde sous chaîne identifie toujours une heure postérieure à celle de la première sous chaîne. Pour permettre les chaînes qui s'étendent au delà de minuit, la valeur de la seconde chaîne peut cependant être inférieure à la valeur de la première sous chaîne. Donc, 'T080000/T210000' identifie la plage de 0800 à 2100, tandis que 'T210000/T080000' identifie la plage de 2100 à 0800 du jour suivant."

" "

"Lorsque une plage s'étend au delà de minuit, elle inclut par définition des parties de deux jours successifs. Lorsque un de ces jours est aussi choisi soit par le MonthOfYearMask, le DayOfMonthMask, et/ou le DayOfWeekMask, mais que l'autre jour ne l'est pas, la politique est alors active seulement durant la portion de la plage qui tombe sur le jour choisi. Par exemple, si la plage s'étend de 2100 à 0800, et si le gabarit de jours de la semaine a choisi les lundis et mardis, la politique est alors active durant les trois intervalles suivants :"

du dimanche minuit au lundi 0800 ; "

du lundi 2100 au mardi 0800 ; "

du mardi 2100 au mardi 23:59:59."

" "

"Si une valeur n'est pas fournie pour cette propriété, la PolicyRule est alors traitée comme valide pour toutes les heures de la journée, et est seulement restreinte par la valeur de sa propriété TimePeriod et les autres propriétés du gabarit.",

ModelCorrespondence {

"CIM_PolicyTimePeriodCondition.TimePeriod",

"CIM_PolicyTimePeriodCondition.LocalOrUtcTime"}

]

Chaîne TimeOfDayMask;

[Description ("Cette propriété indique si les heures représentées dans la propriété TimePeriod et dans les diverses propriétés du gabarit représentent des heures locales ou des temps UTC. Il n'y a pas de disposition pour mélanger les heures locales et les temps UTC : la valeur de cette propriété s'applique à toutes les autres propriétés temporelles."),

ValueMap { "1", "2" },

Valeurs { "localTime", "utcTime" },

ModelCorrespondence {

"CIM_PolicyTimePeriodCondition.TimePeriod",

"CIM_PolicyTimePeriodCondition.MonthOfYearMask",

"CIM_PolicyTimePeriodCondition.DayOfMonthMask",

"CIM_PolicyTimePeriodCondition.DayOfWeekMask",

"CIM_PolicyTimePeriodCondition.TimeOfDayMask"}

]

uint16 LocalOrUtcTime;

};

// =====

// PolicyRuleValidityPeriod

// =====

[Association, Agrégation, Description ("L'agrégation PolicyRuleValidityPeriod représente l'activation et la désactivation programmées d'une PolicyRule. Si une PolicyRule est associée à plusieurs périodes de politique via cette association, la règle est alors active si au moins une des périodes indique qu'elle est active. (En d'autres termes, les PolicyTimePeriodCondition sont OUixées pour déterminer si la règle est active.) Une TimePeriod peut être agrégée par plusieurs PolicyRule. Une règle qui ne pointe pas sur une PolicyTimePeriodCondition via cette association est, du point de vue de la programmation, toujours active. Elle peut cependant être inactive pour d'autres raisons. Par exemple, la propriété activée de la règle peut être réglée à \"désactivé\" (valeur=2).")

]

Classe CIM_PolicyRuleValidityPeriod : CIM_PolicyComponent

{

[Outre passe ("GroupComponent"), Agrège, Description ("Cette propriété contient le nom d'une PolicyRule qui contient une ou plusieurs PolicyTimePeriodCondition.")

]

CIM_PolicyRule REF GroupComponent;

[Outre passe ("PartComponent"), Description ("Cette propriété contient le nom d'une PolicyTimePeriodCondition qui définit les périodes de validité pour une ou plusieurs PolicyRule.")

]

CIM_PolicyTimePeriodCondition REF PartComponent ;

};

```
// =====
// VendorPolicyCondition
// =====
[Description ( "Classe qui fournit un mécanisme général d'extension pour représenter les PolicyCondition qui n'ont pas été
modélisées avec des propriétés spécifiques. À la place, les deux propriétés Constraint et ConstraintEncoding sont utilisées
pour définir le contenu et le format de la condition, comme expliqué ci-dessous."
" "
"Comme son nom le suggère, VendorPolicyCondition est destiné aux extensions spécifiques de fabricant au modèle
d'informations de cœur de politique. Les extensions normalisées ne sont pas supposées utiliser cette classe."
]
Classe CIM_VendorPolicyCondition : CIM_PolicyCondition
{
[Chaîne d'octets, Description ( "Cette propriété fournit un mécanisme général d'extension pour représenter les
PolicyCondition qui n'ont pas été modélisées avec des propriétés spécifiques. Le format de la chaîne d'octets dans le
dispositif est laissé inspecifié dans cette définition. Il est déterminé par la valeur d'OID mémorisée dans la propriété
ConstraintEncoding. Comme ConstraintEncoding est à une seule valeur, toutes les valeurs de Constraint partagent le même
format et la même sémantique."),
ModelCorrespondence {
"CIM_VendorPolicyCondition.ConstraintEncoding"}
]
Chaîne Constraint [];
[Description ( "OID codé comme chaîne, qui identifie le format et la sémantique pour la propriété Constraint de cette
instance."),
ModelCorrespondence {
"CIM_VendorPolicyCondition.Constraint"}
]
Chaîne ConstraintEncoding;
};

// =====
// PolicyAction
// =====
[Abstrait, Description ( "Classe qui représente une action de politique spécifique d'une règle ou réutilisable pour être
effectuée si les PolicyCondition pour une PolicyRule s'évaluent à VRAI. Comme tous les détails du fonctionnement d'une
PolicyAction sont fournis dans les sous classes de cet objet, cette classe est abstraite."
]
Classe CIM_PolicyAction : CIM_Policy
{
[Clé, MaxLen (256), Description ( "Nom de la classe ou sous classe utilisée à la création de l'objet System dans la portée
duquel cette PolicyAction est définie."
" "
"Cette propriété aide à identifier l'objet System dans la portée duquel existe cette instance de PolicyAction. Pour une
PolicyAction spécifique d'une règle, c'est le système dans le contexte duquel la PolicyRule est définie. Pour une
PolicyAction réutilisable, c'est l'instance de PolicyRepository (qui est une sous classe de System) qui détient l'action."
" "
"Noter que cette propriété, et la propriété SystemName analogue, ne représente pas les clés propagées d'une instance de la
classe System. Il y a à la place des propriétés définies dans le contexte de cette classe, qui répètent les valeurs de l'instance
du système auquel se rapporte cette PolicyAction, soit directement via l'agrégation PolicyActionInPolicyRepository, soit
indirectement via l'agrégation PolicyActionInPolicyRule."
]
Chaîne SystemCreationClassName;
[Clé, MaxLen (256), Description ( "Nom de l'objet System dans la portée duquel est définie cette PolicyAction."
" "
"Cette propriété achève l'identification de l'objet System dans la portée duquel existe cette instance de PolicyAction. Pour
une PolicyAction spécifique d'une règle, c'est le système dans le contexte duquel est définie la PolicyRule. Pour une
PolicyAction réutilisable, c'est l'instance de PolicyRepository (qui est une sous classe de System) qui détient l'action."
]
Chaîne SystemName;
[Clé, MaxLen (256), Description ( "Pour une PolicyAction spécifique d'une règle, c'est le CreationClassName de l'objet
PolicyRule auquel cette action est associée. Pour une PolicyAction réutilisable, une valeur spéciale, 'NO RULE', devrait
être utilisée pour indiquer que cette action est réutilisable et non associée à une seule PolicyRule."
]
Chaîne PolicyRuleCreationClassName ;
```

[Clé, MaxLen (256), Description ("Pour une PolicyAction spécifique d'une règle, c'est le nom de l'objet PolicyRule auquel cette action est associée. Pour une PolicyAction réutilisable, une valeur spéciale, 'NO RULE', devrait être utilisée pour indiquer que cette action est réutilisable et non associée à une seule PolicyRule.")

]

Chaîne PolicyRuleName ;

[Clé, MaxLen (256), Description ("CreationClassName indique le nom de la classe ou sous classe utilisée à la création d'une instance. Lorsque utilisée avec les autres propriétés clés de cette classe, cette propriété permet d'identifier de façon univoque toutes les instances de cette classe et de ses sous classes.")]

Chaîne CreationClassName;

[Clé, MaxLen (256), Description ("Nom convivial de cette PolicyAction.")

]

Chaîne PolicyActionName;

};

// =====

// PolicyActionInPolicyRepository

// =====

[Association, Description ("Classe qui représente l'hébergement de PolicyAction réutilisables par un PolicyRepository. Une PolicyAction réutilisable est toujours en rapport avec un seul PolicyRepository, via cette agrégation."

" "

"Noter qu'une instance de PolicyAction peut être réutilisable ou spécifique d'une règle. Lorsque l'action est spécifique d'une règle, elle ne doit pas se rapporter à un PolicyRepository via l'agrégation PolicyActionInPolicyRepository.")

]

Classe CIM_PolicyActionInPolicyRepository : CIM_PolicyInSystem

{

[Outrepassé ("Antécédent"), Max(1), Description ("Cette propriété représente un PolicyRepository qui héberge une ou plusieurs PolicyAction. Une PolicyAction réutilisable se rapporte toujours exactement à un PolicyRepository via l'agrégation PolicyActionInPolicyRepository. La cardinalité [0..1] pour cette propriété couvre les deux types de PolicyAction : 0 pour une PolicyAction spécifique d'une règle, 1 pour une réutilisable.")

]

CIM_PolicyRepository REF Antécédent;

[Outrepassé ("Dépendant"), Description ("Cette propriété détient le nom d'une PolicyAction hébergée dans le PolicyRepository.")

]

CIM_PolicyAction REF Dépendant ;

};

// =====

// PolicyActionInPolicyRule

// =====

[Association, Agrégation, Description ("Une PolicyRule agrège zéro, une ou plusieurs instances de la classe PolicyAction, via l'association PolicyActionInPolicyRule. Une règle qui agrège zéro action n'est pas valide – elle peut cependant être dans un processus d'entrée dans un PolicyRepository ou de définition pour un système. Autrement, les actions de la politique peuvent être explicites dans la définition de la PolicyRule. Noter qu'une PolicyRule devrait n'avoir pas d'effet avant d'être valide."

" "

"Un certain ordre peut être exigé des actions associées à une PolicyRule, ou un ordre recommandé, ou pas d'ordre du tout. Pour les actions représentées comme des objets séparés, l'agrégation PolicyActionInPolicyRule peut être utilisée pour exprimer un ordre."

" "

"Cette agrégation n'indique pas si un ordre d'action spécifié est exigé, recommandé, ou n'a pas de signification ; la propriété SequencedActions dans l'instance agrégante de PolicyRule fournit cette indication."

]

Classe CIM_PolicyActionInPolicyRule : CIM_PolicyComponent

{

[Outrepassé ("GroupComponent"), Agrège, Description ("Cette propriété représente la PolicyRule qui contient une ou plusieurs PolicyAction.")

]

CIM_PolicyRule REF GroupComponent;

[Outrepassé ("PartComponent"), Description ("Cette propriété détient le nom d'une PolicyAction contenue par une ou plusieurs PolicyRule.")

]

CIM_PolicyAction REF PartComponent;

[Description ("Cette propriété donne un entier non signé 'n' qui indique la position relative d'une PolicyAction dans la séquence des actions associées à une PolicyRule. Lorsque 'n' est un entier positif, il indique une place dans la séquence des actions à effectuer, les plus petits entiers indiquant les premières positions dans la séquence. La valeur spéciale '0' indique 'ne pas tenir compte'. Si deux PolicyAction ou plus ont le même numéro de séquence non zéro, elles peuvent être effectuées dans n'importe quel ordre, mais elles doivent toutes être effectuées à la place appropriée dans la séquence d'actions globale."
" "

"Une série d'exemples va préciser l'ordre des PolicyAction :"

o Si toutes les actions ont le même numéro de séquence, sans considération de si c'est '0' ou non, tout ordre est acceptable."

o Les valeurs :

1 : ACTION A

2 : ACTION B

1 : ACTION C

3 : ACTION D

indiquent deux ordres acceptables : A,C,B,D ou C,A,B,D, car A et C peuvent être effectués dans l'un ou l'autre ordre, mais seulement à la position '1'.

o Les valeurs :"

0 : ACTION A

2 : ACTION B

3 : ACTION C

3 : ACTION D

exigent que B,C, et D se produisent soit comme B,C,D, soit comme B,D,C. L'action A peut apparaître en tout point par rapport à B, C, et D. Donc, l'ensemble complet d'ordres acceptables est : A,B,C,D ; B,A,C,D ; B,C,A,D ; B,C,D,A ; A,B,D,C ; B,A,D,C ; B,D,A,C ; B,D,C,A.

" "

"Noter que les numéros de séquence non à zéro n'ont pas besoin de commencer à '1', et il n'est pas nécessaire qu'ils soient consécutifs. Tout ce qui importe est leur magnitude relative."

```
]
uint16 ActionOrder;
};
```

```
// =====
// VendorPolicyAction
// =====
```

[Description ("Classe qui fournit un mécanisme général d'extension pour représenter les PolicyAction qui n'ont pas été modélisées avec des propriétés spécifiques. À la place, les deux propriétés ActionData et ActionEncoding sont utilisées pour définir le contenu et le format de l'action, comme expliqué ci-dessous."
" "

"Comme son nom le suggère, VendorPolicyAction est destiné aux extensions spécifiques de fabricant au modèle d'informations de cœur de politique. Des extensions normalisées ne sont pas supposées utiliser cette classe.")]

Classe CIM_VendorPolicyAction : CIM_PolicyAction

```
{
```

[Chaîne d'octets, Description ("Cette propriété donne un mécanisme général d'extension pour représenter des PolicyAction qui n'ont pas été modélisées avec des propriétés spécifiques. Le format des chaînes d'octets dans le dispositif est laissé inspecifié dans cette définition. Il est déterminé par la valeur d'OID mémorisée dans la propriété ActionEncoding. Comme ActionEncoding est à une seule valeur, toutes les valeurs de ActionData partagent le même format et la même sémantique."),

ModelCorrespondence {"CIM_VendorPolicyAction.ActionEncoding"}
]

Chaîne ActionData [];

[Description ("OID codé comme une chaîne, qui identifie le format et la sémantique pour la propriété ActionData de cette instance."),

ModelCorrespondence {

"CIM_VendorPolicyAction.ActionData"}
]

Chaîne ActionEncoding ;

```
};
```

```
// =====
// Fin de fichier
// =====
```

15. Déclaration complète de droits de reproduction

Copyright (C) The Internet Society (2001). Tous droits réservés.

Ce document et les traductions de celui-ci peuvent être copiés et diffusés, et les travaux dérivés qui commentent ou expliquent autrement ou aident à sa mise en œuvre peuvent être préparés, copiés, publiés et distribués, partiellement ou en totalité, sans restriction d'aucune sorte, à condition que l'avis de copyright ci-dessus et ce paragraphe soit inclus sur toutes ces copies et œuvres dérivées. Toutefois, ce document lui-même ne peut être modifié en aucune façon, par exemple en supprimant le droit d'auteur ou les références à l'Internet Society ou d'autres organisations Internet, sauf si c'est nécessaire à l'élaboration des normes Internet, auquel cas les procédures pour les droits de reproduction définis dans les processus des normes pour l'Internet doivent être suivies, ou si nécessaire pour le traduire dans des langues autres que l'anglais.

Les permissions limitées accordées ci-dessus sont perpétuelles et ne seront pas révoquées par la Société Internet, ses successeurs ou ayants droit.

Ce document et les renseignements qu'il contient sont fournis "TELS QUELS" et l'INTERNET SOCIETY et l'INTERNET ENGINEERING TASK FORCE déclinent toute garantie, expresse ou implicite, y compris mais sans s'y limiter, toute garantie que l'utilisation de l'information ici présente n'enfreindra aucun droit ou aucune garantie implicite de commercialisation ou d'adaptation à un objet particulier.

Remerciement

Le financement de la fonction d'éditeur des RFC est actuellement fourni par la Internet Society.