

Groupe de travail Réseau

**Request for Comments : 2533**

Catégorie : En cours de normalisation

G. Klyne, Content Technologies/5GM

mars 1999

Traduction : Clauce Brière de L'Isle

## Syntaxe pour la description des ensembles de caractéristiques de support

### Statut du présent Mémo

La présente RFC spécifie un protocole de normalisation pour la communauté Internet et appelle à des discussions et suggestions pour son amélioration. Prière de se reporter à l'édition en cours des "Internet Official Protocol Standards" (*normes officielles du protocole Internet*) (STD 1) pour connaître l'état de la normalisation et le statut du présent protocole. La distribution du présent mémo n'est soumise à aucune restriction.

(Version corrigée conformément à la RFC2738, aux paragraphes 4.1 et 5.8.2)

### Déclaration de Copyright

Copyright (C) The Internet Society (1999). Tous droits réservés.

### Résumé

Un certain nombre de protocoles d'application de l'Internet ont besoin de fournir une négociation du contenu pour les ressources avec laquelle ils interagissent [1]. Un cadre pour de telles négociations est décrit dans [2], dont une partie est une façon de décrire la gamme de caractéristiques de support qui peuvent être traitées par l'expéditeur, le receveur ou par le format de transmission de document d'un message. Un format pour un vocabulaire des caractéristiques individuelles des supports et les procédures pour l'enregistrement des caractéristiques sont présentés dans [3].

Le présent document introduit et décrit une syntaxe qui peut être utilisée pour définir des ensembles de caractéristiques qui sont formées à partir de combinaisons et relations qui impliquent des caractéristiques de support individuelles. De tels ensembles de caractéristiques sont utilisés pour décrire les capacités de traitement des caractéristiques de support des expéditeurs et receveurs de message et des formats de fichier.

Un algorithme pour l'établissement des correspondances d'ensembles de caractéristiques est également décrit.

### Table des Matières

1.	Introduction.....
1.1	Structure du document.....
1.2	Terminologie et conventions du document.....
1.3	Discussion du document.....
2.	Terminologie et définitions des caractéristiques de support.....
3.	Combinaisons et capacités de caractéristiques de support.....
3.1	Caractéristiques de support.....
3.2	Collections et ensembles de caractéristiques de support.....
3.3	Descriptions d'ensemble de caractéristiques de support.....
3.4	Scénario de combinaison de caractéristiques de support.....
3.5	Prédicats d'ensemble de caractéristiques.....
3.6	Description des préférences.....
3.7	Combinaison des préférences.....
4.	Représentation d'ensemble de caractéristiques.....
4.1	Représentation textuelle des prédicats.....
4.2	Interprétation de la syntaxe de prédicat de caractéristique.....
4.3	Exemple de définition d'ensemble de caractéristiques.....
5.	Correspondance des ensembles de caractéristiques.....
5.1	Stratégie de confrontation d'ensembles de caractéristiques.....
5.2	Formulation du prédicat cible.....
5.3	Remplacer les expressions de "set".....
5.4	Déplacer vers l'intérieur les négations logiques.....
5.5	Remplacer les comparaisons et négations logiques.....
5.6	Conversion en forme canonique.....
5.7	Groupement des prédicats de caractéristiques.....
5.8	Fusion des contraintes de caractéristiques seules.....
6.	Autres caractéristiques et questions.....
6.1	Prédicats désignés et auxiliaires.....
6.2	Désignation des unités.....

6.3	<a href="#">Types inconnus de données de valeur de caractéristique.....</a>
7.	<a href="#">Exemples et commentaires supplémentaires.....</a>
7.1	<a href="#">Exemple élaboré.....</a>
7.2	<a href="#">Note sur la portée d'une étiquette de caractéristique.....</a>
8.	<a href="#">Considérations pour la sécurité.....</a>
9.	<a href="#">Remerciements.....</a>
10.	<a href="#">Références.....</a>
11.	<a href="#">Adresse de l'auteur.....</a>
	<a href="#">Déclaration complète des droits de reproduction.....</a>

## 1. Introduction

Un certain nombre de protocoles d'application de l'Internet ont besoin de fournir une négociation de contenu pour les ressources avec lesquelles ils interagissent [1]. Un cadre pour de telles négociations est décrit dans [2]. Une partie de ce cadre est le moyen de décrire la gamme des caractéristiques des supports qui peuvent être traitées par l'expéditeur, le receveur ou le format de transmission de document d'un message.

Les descriptions des capacités en caractéristiques de support doivent se fonder sur un vocabulaire sous-jacent de caractéristiques individuelles de support. Un format pour un tel vocabulaire et les procédures pour l'enregistrement des caractéristiques de support au sein de ce vocabulaire sont présentés dans [3].

Le présent document définit une syntaxe qui peut être utilisée pour décrire les ensembles de caractéristiques qui sont formés à partir des combinaisons et des relations qui impliquent les caractéristiques de support individuelles. De tels ensembles de caractéristiques sont utilisés pour décrire les capacités de traitement du support par les expéditeurs et receveurs de message et les formats de fichier.

Un algorithme pour établir la correspondance des ensembles de caractéristiques est également décrit ici.

La syntaxe d'ensemble de caractéristique est construite sur le principe d'utilisation des prédicats des ensembles de caractéristiques comme des "relations mathématiques" qui définissent des contraintes sur les capacités de traitement d'une caractéristique. Cela permet que la même forme d'expression d'ensemble de caractéristique soit utilisée pour décrire les capacités de l'expéditeur, du receveur et le format du fichier. Cela a été modélisé de façon libre en ce sens que les bases de données relationnelles utilisent des expressions booléennes pour décrire un ensemble de valeurs de résultat, et une syntaxe qui se fonde sur les filtres de recherche de LDAP.

### 1.1 Structure du document

La partie principale de ce mémoire traite des domaines suivants :

La Section 2 introduit des termes qui sont utilisés avec une signification particulière.

La Section 3 introduit le concept de description des capacités de traitement des supports comme de combinaisons de caractéristiques de support possibles, et l'idée d'utiliser des expressions booléennes pour exprimer de telles combinaisons.

La Section 4 contient une description d'une syntaxe pour la description des ensembles de caractéristiques fondée sur l'idée introduite précédemment d'utiliser des expressions booléennes pour décrire les combinaisons de caractéristiques de support.

La Section 5 décrit un algorithme pour établir la correspondance des ensembles de caractéristiques.

La Section 6 expose quelques questions supplémentaires de description et de traitement de caractéristique de support qui peuvent être vues comme des extensions du cadre central.

La Section 7 contient un exemple élaboré de mise en correspondance d'ensembles de caractéristique et quelques commentaires explicatifs supplémentaires provoqués par les questions soulevées par l'application de ce cadre aux transmissions de télécopie.

## 1.2 Terminologie et conventions du document

Les mots clés "DOIT", "NE DOIT PAS", "EXIGE", "DEVRA", "NE DEVRA PAS", "DEVRAIT", "NE DEVRAIT PAS", "RECOMMANDE", "PEUT", et "FACULTATIF" dans ce document sont à interpréter comme décrit dans la RFC2119.

Note : Des commentaires comme celui-ci apportent des informations supplémentaires non essentielles sur les raisons qui servent de base au document. De telles informations ne sont pas nécessaires pour construire une mise en œuvre conforme, mais peuvent aider celui qui souhaite accéder à une compréhension plus approfondie des concepts.

*NDT (Tout ce paragraphe est un exemple de l'humour britannique fondé sur le non sens ; aucune occurrence de ces "mots clés" ne figure dans le texte, et le commentaire de la note n'explique rien sur la présence de ces mots clés.)*

## 1.3 Discussion du document

La discussion de ce document devrait avoir lieu dans le cadre de la liste de diffusion sur la négociation du contenu et l'enregistrement des caractéristiques du support hébergée par le consortium de la messagerie Internet (IMC, *Internet Mail Consortium*).

Prière d'envoyer les commentaires concernant ce document à [ietf-medfree@imc.org](mailto:ietf-medfree@imc.org)

Pour s'abonner à cette liste, envoyer un message avec le corps de message "subscribe" à [ietf-medfree-request@imc.org](mailto:ietf-medfree-request@imc.org).

Pour voir ce qui s'est passé avant votre abonnement, prière de se reporter aux archives de la liste de diffusion à <http://www.imc.org/ietf-medfree/>

## 2. Terminologie et définitions des caractéristiques de support

Collection de caractéristiques

C'est une collection de différentes caractéristiques de support et des valeurs associées. Cela peut être vu comme la description d'un rendu spécifique ou une instance spécifique d'un document ou ressource par un receveur spécifique.

Ensemble de caractéristiques

C'est un ensemble de zéro, une ou plusieurs collections de caractéristiques.

Note : Ce terme était utilisé de façon légèrement différente par les travaux antérieurs sur la négociation transparente de contenu dans HTTP [4].

Prédicat d'ensemble de caractéristiques

C'est une fonction d'une valeur arbitraire de collection de caractéristiques qui retourne un résultat booléen. Un résultat VRAI signifie que la collection de caractéristiques correspondante appartient à un certain ensemble de capacités de traitement de caractéristiques de support définies par ce prédicat.

Les autres termes utilisés dans le présent mémoire sont définis dans [2].

## 3. Combinaisons et capacités de caractéristiques de support

### 3.1 Caractéristiques de support

Le présent mémoire suppose que les valeurs individuelles des caractéristiques de support sont de simples valeurs atomiques :

- o valeurs booléennes,
- o valeur énumérées,
- o valeur de chaînes de texte (traitée comme des entités atomiques, comme des jetons de valeur énumérés),
- o valeurs numériques (entières ou rationnelles).

Ces valeurs ont toutes la propriété de pouvoir être comparées pour égalité ('='), et que les valeurs numériques et d'énumération ordonnée peuvent être comparées dans des relations de plus grand que ou plus petit que ('<', '>'). Ces opérations de comparaison de base sont utilisées comme blocs de construction de primitives pour des expressions de capacité plus générales.

### 3.2 Collections et ensembles de caractéristiques de support

Une seule valeur quelconque de caractéristique de support peut être vue comme un composant d'une collection de caractéristiques qui décrit une instance d'une ressource (par exemple, un document imprimé, une image affichée, etc.). Une telle collection de caractéristiques consiste en un certain nombre d'étiquettes de caractéristiques de support (chacune selon [3]) et les valeurs de caractéristiques associées.

Un ensemble de caractéristiques contient un certain nombre de collections de caractéristiques. Donc, un ensemble de caractéristiques peut décrire un certain nombre de différentes instances de ressources de données. Cela peut correspondre à différents traitements d'une seule ressource de données (par exemple, différentes résolutions utilisées pour imprimer un document donné) un certain nombre de différentes ressources de données soumises à un traitement commun (par exemple, la gamme des différentes images qui peuvent être rendues sur un affichage donné) ou quelque combinaison d'entre elles (voir les exemples ci-dessous).

Donc, une description d'un ensemble de caractéristiques peut décrire les capacités d'une ressource de données ou une entité qui traite ou restitue une ressource de données.

### 3.3 Descriptions d'ensemble de caractéristiques de support

Un ensemble de caractéristiques peut être non bordé. Par exemple, en principe, il n'y a pas de limite au nombre de différents documents qui peuvent être sortis en utilisant une imprimante donnée. Mais pour être utilisable en pratique, une description d'ensemble de caractéristiques doit être finie.

L'approche générale pour décrire les ensembles de caractéristiques est de commencer à partir de l'hypothèse que tout est possible ; c'est-à-dire que l'ensemble de caractéristiques contient toutes les instances possibles de documents (les collections de caractéristiques). Des contraintes sont alors appliquées qui retirent progressivement les instances de document de cet ensemble ; par exemple, pour une imprimante monochrome, toutes les instances de document qui utilisent la couleur sont retirées, ou pour un document qui doit être rendu à une résolution minimale, toutes les instances de document ayant des résolutions inférieures sont retirées de l'ensemble. Le mécanisme utilisé pour retirer les instances de document de l'ensemble est l'idée mathématique de "relation" ; c'est-à-dire une fonction booléenne (un "prédicat") qui prend un paramètre de collection de caractéristiques et retourne une valeur booléenne qui est VRAI si la collection de caractéristiques décrit une instance acceptable de document, ou FAUX si elle en décrit une qui est exclue.

```

                P (C)
P (C) = VRAI <- : -> P (C) = FAUX
                :
+-----+-----+ Cette boîte représente des ensembles
|         :         | de collections de caractéristiques (C)
| Inclus  : Exclus  | qui sont contraints par le prédicat P.
|         :         |
+-----+-----+
                :
```

Le résultat de l'application d'une série de telles contraintes est un plus petit ensemble de collections de caractéristiques qui représente une certaine capacité de traitement du support. Lorsque les contraintes individuelles sont représentées par des prédicats qui décrivent chacun une capacité de traitement du support, les effets combinés de ces contraintes sont un sous-ensemble de la capacité individuelle contrainte qui peut être représentée par un prédicat qui est le ET logique des prédicats des contraintes individuelles.

### 3.4 Scénario de combinaison de caractéristiques de support

Ce paragraphe développe des exemples de scénarios, et introduit la notation dont la définition formelle est à la section 4.

#### 3.4.1 Options de ressource de données

L'expression suivante décrit une ressource de données qui peut être affichée comme :

- (a) une image de 750 x 500 pixels utilisant 15 couleurs, ou
- (b) à 150 dpi sur une page A4.

```
(| (& (pix-x=750) (pix-y=500) (color=15) )
 (& (dpi>=150) (papersize=iso-A4) ) )
```

### 3.4.2 Capacités du receveur

L'expression suivante décrit un système receveur qui a :

- (a) un écran capable d'afficher 640\*480 pixels et 16 millions de couleurs (24 bits par pixel), 800\*600 pixels et 64 000 couleurs (16 bits par pixel) ou 1024\*768 pixels et 256 couleurs (8 bits par pixel), ou
- (b) une imprimante capable de rendre 300dpi en papier A4.

```
(| (& (| (& (pix-x<=640) (pix-y<=480) (color<=16777216) )
 (& (pix-x<=800) (pix-y<=600) (color<=65535) )
 (& (pix-x<=1024) (pix-y<=768) (color<=256) ) )
 (ua-media=screen) )
 (& (dpi=300)
 (ua-media=stationery) (papersize=iso-A4) ) )
```

Noter que cette expression ne dit rien sur les capacités de couleur ou d'échelle de gris de l'imprimante. Dans le schéma présenté ici, on suppose qu'il n'y a pas de contrainte à cet égard (ou, de façon plus réaliste, que de telles contraintes sont traitées hors bande par ceux qui envoient à ce receveur).

### 3.4.3 Options combinées

L'exemple suivant décrit la gamme de représentations de document disponibles lorsque la ressource décrite dans le premier exemple ci dessus est envoyée au receveur décrit dans le second exemple. C'est le résultat de la combinaison de leurs ensembles de caractéristiques de capacités :

```
(| (& (pix-x=750) (pix-y=500) (color=15) )
 (& (dpi=300) (ua-media=stationery) (papersize=iso-A4) ) )
```

L'ensemble de caractéristiques décrit par cette expression est l'intersection des ensembles décrits par les deux expressions de capacités précédentes.

## 3.5 Prédicats d'ensemble de caractéristiques

Il y a de nombreuses façons de représenter un prédicat. Les idées exposées dans le présent mémoire ont été inspirées par le langage de programmation Prolog [5], et son utilisation des prédicats pour décrire des ensembles d'objets.

Pour les besoins de la description des caractéristiques de supports dans les protocoles d'application de réseautage, le format utilisé par les filtres de recherche LDAP [7], [8] a été adopté, parce qu'il correspond bien aux exigences de l'identification des capacités, et a une structure très simple qui est facile à analyser et à traiter.

### 3.5.1 Comparaison avec les filtres de recherche dans les répertoires

On observe qu'une collection de caractéristiques est similaire à une entrée de répertoire, en ce qu'elle consiste en une collection de valeurs désignées. De plus, la sémantique du mécanisme de choix des collections de caractéristiques à partir d'un ensemble de caractéristiques est à de nombreux égards similaire à la sélection d'entrées d'un répertoire.

Un prédicat d'ensemble de caractéristiques utilisé pour décrire les capacités de traitement d'un support est implicitement appliqué à une collection de caractéristiques. Au sein du prédicat, les membres de la collection de caractéristiques sont identifiés par leurs étiquettes de caractéristique, et sont comparés avec les valeurs de caractéristiques connues. (À comparer avec la façon dont un filtre de recherche LDAP est appliqué à une entrée de répertoire, dont les membres sont identifiés par les noms de type d'attribut, et comparés aux valeurs d'attribut connues.)

Par exemple, dans (& (dpi>=150) (papersize=iso-A4) ) les jetons 'dpi' et 'papersize' sont des étiquettes de caractéristiques, et '150' et 'iso-A4' sont des valeurs de caractéristiques. (Dans le filtre de recherche LDAP correspondant, il y aurait un type d'attribut d'entrée de répertoire et des valeurs d'attribut.)

Les différences entre la sélection de répertoire (selon [7]) et la sélection d'un ensemble de caractéristiques sont :

- o La sélection de répertoire donne des correspondances de sous chaînes, d'approximations, et d'extension pour les valeurs d'attribut. Une telle correspondance n'est pas fournie pour la sélection d'ensemble de caractéristiques.
- o La sélection de répertoire peut être fondée sur la présence d'un attribut sans égard à sa valeur. Dans le cadre sémantique décrit par ce document, les essais de caractéristiques de valeurs booléennes peuvent être utilisés pour donner un effet similaire.
- o La sélection de répertoire donne des règles de correspondance qui vérifient la présence ou l'absence d'un type d'attribut désigné.
- o La sélection de répertoire donne des règles de correspondance qui dépendent du type de données déclaré d'une valeur d'attribut.
- o La sélection de caractéristique donne l'association d'une valeur de qualité avec un prédicat de caractéristique comme moyen de classer les collections de valeurs choisies.

Dans le cadre sémantique décrit dans ce document, les essais de caractéristiques à valeurs booléennes peuvent être utilisées lorsque des essais de présence seraient utilisés dans un filtre de recherche de répertoire.

L'idée d'une correspondance et de règles de correspondance extensibles selon les types de données sont les facettes d'un problème qui n'est pas traité par le présent mémoire, mais qui n'affecte pas nécessairement la syntaxe de sélection de caractéristiques. Un aspect qui peut peser sur la syntaxe serait la spécification d'une règle explicite de règle de correspondance au titre d'une expression de sélection.

### 3.6 Description des préférences

Un façon pratique pour décrire les préférences est d'utiliser des "valeurs de qualité" numériques.

Il a été suggéré que les valeurs de qualité numériques seraient potentiellement trompeuses si elles étaient utilisées comme plus qu'une simple façon de classer les options. Pour les besoins du présent mémoire, le classement des options est suffisant.

On utilise les valeurs de qualité numériques dans la gamme de 0 à 1, avec jusqu'à trois chiffres après la virgule pour classer les ensembles de caractéristiques selon la préférence. Les plus fortes valeurs sont préférées aux valeurs inférieures, et les valeurs égales sont supposées de préférence égale. Au delà de cela, le nombre réel utilisé n'a pas de signification définie ici. Les opérations arithmétiques sur les valeurs de qualité vont vraisemblablement produire des résultats imprévisibles si une sémantique appropriée n'a pas été définie pour le contexte dans lequel de telles opérations sont utilisées.

En l'absence d'une valeur de qualité explicitement appliquée, une valeur de "1" est supposée.

En utilisant la notation définie plus loin, une valeur de qualité peut être attribuée à toute sous-expression de prédicat d'ensemble de caractéristiques :

```
(| (& (pix-x=750) (pix-y=500) (color=15) );q=0.8
  (& (dpi>=150) (papersize=iso-A4) ) ;q=0.7 )
```

Le paragraphe 3.7 ci-dessous explique que les valeurs de qualité attribuées aux sous-expressions ne sont pas toujours utiles.

Note : La syntaxe des valeurs de qualité utilisée ici est tirée de celle définie pour les en-têtes HTTP 'Accept:' au paragraphe 3.9 de la RFC2068 [9]. Cependant, l'utilisation des valeurs de qualité définies ici ne va pas aussi loin que celles définies dans la RFC2068.

### 3.7 Combinaison des préférences

Le problème général de la description et de la combinaison des préférences parmi les ensembles de caractéristiques est beaucoup plus complexe que de simplement décrire les ensembles de caractéristiques admissibles. Par exemple, étant donné deux ensembles de caractéristiques :

```
(& (a1);q=0.8 (b1);q=0.7 )
```

(& (a2);q=0.5 (b2);q=0.9 )

où :

la caractéristique a1 est préférée à a2

la caractéristique b2 est préférée à b1

Lequel de ces ensembles de caractéristiques est préféré ? En l'absence d'informations ou hypothèses supplémentaires, il n'y a pas de réponse généralement satisfaisante à cette question.

La solution proposée à cette question est simplement de dire qu'aucune règle n'est fournie pour combiner les informations de préférence. Appliquée à l'exemple ci-dessus, toute information de préférence sur (a1) en relation avec (a2), ou (b1) en relation avec (b2) n'est pas supposée apporter d'informations sur la préférence de (& (a1) (b1) ) en relation avec (& (a2) (b2) ).

En termes pratiques, cela restreint l'application des informations de préférence aux clauses de prédicat de niveau supérieur. Une clause de niveau supérieur définit complètement un ensemble de caractéristiques admissible ; les clauses combinées par les opérateurs ET logiques ne peuvent pas être des clauses de niveau supérieur (voir le format canonique pour les prédicats d'ensemble de caractéristiques, décrit plus loin).

Note : Le présent mémoire ne donne pas une signification spécifique aux valeurs de qualité ou aux règles pour les combiner. L'application de telles significations et règles n'est pas interdite, mais est vue comme une domaine où se poursuivent les recherches et l'expérimentation. Un exemple d'un concept qui utilise une sémantique de valeur de qualité étendue et d'opération de combinaison est "Négociation de contenu transparent dans HTTP" [4]. D'autres travaux qui étendent aussi les valeurs de qualité portent sur l'algorithme de négociation de contenu dans le serveur Apache HTTP [14].

## 4. Représentation d'ensemble de caractéristiques

Les sections précédentes ont décrit un cadre pour la définition des ensembles de caractéristiques avec des prédicats appliqués aux collections de caractéristiques. La présente section présente une représentation concrète des prédicats d'ensemble de caractéristiques.

### 4.1 Représentation textuelle des prédicats

La représentation textuelle d'un ensemble de caractéristiques se fonde sur la RFC2254 "Représentation comme chaîne des filtres de recherche LDAP" [8], en excluant les éléments qui ne sont pas pertinents pour le choix des ensembles de caractéristiques (exposé plus haut) et en ajoutant des éléments spécifiques du choix des ensembles de caractéristiques (par exemple, les options pour associer valeurs de qualité et prédicats).

Le format d'un prédicat de caractéristique est défini par les productions pour "filtre" dans ce qui suit, en utilisant la notation syntaxique et les règles clés de la RFC2234 [10] :

filtre	= "(" filtercomp ")" *( ";" paramètre )
paramètre	= "q" "=" qvalue / ext-param "=" ext-value
qvalue	= ("0" [ "." 0*3DIGIT ] ) / ("1" [ "." 0*3("0") ] )
ext-param	= ALPHA *( ALPHA / DIGIT / "-" )
ext-value	= <valeur du paramètre, conformément au paramètre désigné>
filtercomp	= et / ou / non / item
et	= "&" filterlist
ou	= " " filterlist
non	= "!" filtre
filterlist	= 1*filtre
item	= simple / set / ext-pred
set	= attr "=" "[" setentry *( ";" setentry ) "]"
setentry	= valeur / gamme
gamme	= valeur "." valeur
simple	= attr filtertype valeur
filtertype	= égal / supérieur / inférieur
égal	= "="
supérieur	= ">="

inférieur	= " $\leq$ "
attr	= fanion
valeur	= fvalue
fanion	= <étiquette de caractéristique, comme définie dans la RFC2506 [3]>
fvalue	= Booléen / nombre / jeton / chaîne
Booléen	= "VRAI" / "FAUX"
nombre	= entier / rationnel
entier	= [ "+" / "-" ] 1*CHIFFRE
rationnel	= [ "+" / "-" ] 1*CHIFFRE "/" 1*CHIFFRE
jeton	= ALPHA *( ALPHA / CHIFFRE / "-" )
chaîne	= DQUOTE *(%x20-21 / %x23-7E) DQUOTE ; chaîne entre guillemets de SP et de VCHAR sans DQUOTE
ext-pred	= <Prédicat d'extension de contrainte, non défini ici>

(Sous réserve des contraintes imposées par le protocole qui porte un prédicat de caractéristique, les caractères d'espace peuvent apparaître entre toute paire d'éléments syntaxiques ou de littéraux qui apparaissent à la droite de ces productions.)

Comme il est décrit, la syntaxe permet que des paramètres (y compris des valeurs de qualité) soient attachés à toute valeur de "filtre" dans le prédicat (et pas seulement des valeurs de niveau supérieur). Seules les valeurs de qualité de niveau supérieur sont reconnues. Si aucune valeur de qualité explicite n'est donnée, une valeur de '1,0' est appliquée.

Note : L'approche souple des valeurs de qualité et des autres valeurs de paramètre dans la syntaxe a été adoptée pour deux raisons : (a) pour faciliter la combinaison des prédicats de caractéristiques construits séparément, et (b) pour fournir un mécanisme extensible d'étiquetage pour d'éventuelles utilisations futures (par exemple, pour incorporer une exigence concevable de spécification explicite de règle de correspondance).

## 4.2 Interprétation de la syntaxe de prédicat de caractéristique

Un prédicat d'ensemble de caractéristiques est décrit par la production syntaxique de "filtre".

### 4.2.1 Syntaxe de filtre

Un "filtre" est défini soit comme une simple comparaison de caractéristiques ('item', voir ci-dessous) soit comme un filtre composite ("et", "ou", "non"), habillé de valeurs de paramètres facultatifs (y compris "q=qvalue").

Un filtre composite est une combinaison logique d'une ou plusieurs valeurs de "filtre" : (& f1 f2 ... fn ) est le ET logique des valeurs de filtre "f1", "f2" à "fn". C'est à dire qu'il est satisfait par toute collection de caractéristiques qui satisfait tous les prédicats représentés par ces filtres.

(| f1 f2 ... fn ) est le OU logique des fvaleurs de filtre "f1", "f2" à "fn". C'est à dire qu'il est satisfait par toute collection de caractéristiques qui satisfait au moins un des prédicats représentés par ces filtres.

(! f1 ) est la négation logique de la valeur de filtre "f1". C'est à dire qu'elle est satisfaite par toute collection de caractéristiques qui NE satisfait PAS le prédicat représenté par "f1".

### 4.2.2 Comparaison de caractéristiques

Une comparaison de caractéristiques est définie par l'option "simple" de la production syntaxique pour "item". Il y a trois formes de base :

(fanion=valeur) compare la caractéristique nommée "fanion" (dans certaines collections de caractéristiques qui font l'objet des essais) avec la "valeur" fournie pour voir si elles sont égales. Cela peut être utilisé avec tout type de valeur de caractéristique (numérique, booléenne, jeton ou chaîne).

(fanion≤valeur) compare la caractéristique numérique nommée "fanion" avec la "valeur" fournie pour voir si la caractéristique est inférieure ou égale à la "valeur".

(fanion≥valeur) compare la caractéristique numérique nommée "fanion" avec la "valeur" fournie pour voir si la caractéristique est supérieure ou égale à la "valeur".

Les essais de "supérieur à" et "inférieur à" peuvent être effectués avec des valeurs de caractéristiques qui ne sont pas numériques, mais en général, ils reviennent à des essais d'égalité car il n'y a pas de relation d'ordre sur les valeurs non

numériques définies par la présente spécification. Des applications spécifiques peuvent définir de telles relations d'ordre sur des étiquettes de caractéristiques spécifiques, mais de telles définitions sortent du domaine d'application de la présente spécification (qui n'exige pas leur conformité).

### 4.2.3 Étiquettes de caractéristiques

Les étiquettes de caractéristiques se conforment à la syntaxe donnée dans "Procédure d'enregistrement d'étiquette de caractéristique de support" [3]. Les étiquettes de caractéristiques utilisées pour décrire des capacités devraient être enregistrées en utilisant les procédures décrites dans le présent mémoire. Les étiquettes de caractéristiques non enregistrées devraient être allouées dans "l'arborescence d'URI", comme exposé dans le mémoire sur les procédures d'enregistrement des caractéristiques des supports [3].

Si une étiquette de caractéristiques non reconnue est rencontrée dans le cours du traitement d'un prédicat d'ensemble de caractéristiques, elle devrait quand même être traitée comme une étiquette de caractéristique légitime. Les règles de correspondance d'ensembles de caractéristiques sont conçues pour permettre que soient introduites de nouvelles étiquettes de caractéristiques sans affecter la validité des assertions de capacités existantes.

### 4.2.4 Valeurs de caractéristiques

Une caractéristique peut avoir une valeur numérique, booléenne, de jeton ou de chaîne.

#### 4.2.4.1 Valeurs booléennes

Un booléen est simplement un jeton avec deux valeurs prédéfinies : "VRAI" et "FAUX". (Les lettres majuscules ou minuscules peuvent être utilisées sous toute combinaison.)

#### 4.2.4.2 Valeurs numériques

Une valeur numérique est soit un entier décimal, facultativement précédé d'un signe "+" ou "-", soit un nombre rationnel.

Un nombre rationnel est exprimé par "n/m", facultativement précédé d'un signe "+" ou "-". Le "n" et le "m" sont des entiers décimaux non signés, et la valeur représentée par "n/m" est "n" divisé par "m". Donc, les représentations du nombre 1,5 ci-après sont toutes valides : 3/2 ; +15/10 ; 600/400.

Donc, plusieurs formes de nombre rationnel peuvent exprimer la même valeur. Une forme canonique de nombre rationnel est obtenue en trouvant le plus grand facteur commun de "n" et "m", et en divisant "n" et "m" tous deux par cette valeur.

Une simple valeur d'entier peut être utilisée partout à la place d'un nombre rationnel. Nous avons donc :

+ 5 est équivalent à + 5/1 ou + 50/10, etc.

- 2 est équivalent à - 2/1 ou - 4/2, etc.

Tout signe dans un nombre rationnel doit précéder le nombre entier, de sorte que les nombres suivants ne sont pas des rationnels valides :

3/+2, 15/-10 (\*\*NON VALIDES\*\*)

#### 4.2.4.3 Valeurs de jeton

Une valeur de jeton est toute séquence de lettres, chiffres et de caractères "-" qui se conforme à la syntaxe pour "jeton" donnée ci-dessus. C'est un nom donné pour une certaine valeur (non spécifiée).

#### 4.2.4.4 Valeurs de chaîne

Une valeur de chaîne est toute séquence de caractères incluse entre des guillemets et qui se conforme à la syntaxe donnée ci-dessus pour "chaîne".

La sémantique de chaîne définie par le présent mémoire est la même que celle d'une valeur de jeton. Mais une chaîne permet une bien plus grande variété de formats internes, et des applications spécifiques peuvent choisir d'interpréter le contenu de façons qui vont au delà de celles données ici. Lorsque une telle interprétation est possible, les formats de chaînes permis et les interprétations correspondantes devraient être indiquées dans l'enregistrement de caractéristique de support (selon la RFC2506 [3]).

### 4.2.5 Facilités de notation

L'option 'set' de la production syntaxique pour 'item' est simplement un raccourci de notation pour certaines situations

courantes qui peuvent être exprimées en utilisant des constructions "simples". Les occurrences d'éléments 'set' peuvent être éliminées en appliquant les identités suivantes :

$$\begin{aligned} T = [ E1, E2, \dots, En ] & \rightarrow ( ( T=[E1] ) ( T=[E2] ) \dots ( T=[En] ) ) \\ ( T=[R1..R2] ) & \rightarrow ( \& ( T \geq R1 ) ( T \leq R2 ) ) \\ ( T=[E] ) & \rightarrow ( T=E ) \end{aligned}$$

Exemples :

L'expression ( paper-size=[A4,B4] ) (*taille de papier*) peut être utilisée pour exprimer une capacité à imprimer des documents sur du papier de taille aussi bien A4 que B4.

L'expression ( width=[4..17/2] ) (*largeur*) peut être utilisée pour exprimer une capacité à imprimer des documents dont la largeur est comprise entre 4 et 8,5 pouces. (*Voir cependant le paragraphe 2.3 de la RFC2506*)

La construction "set" est conçue pour que les valeurs et gammes énumérées puissent être combinées en une seule expression, par exemple : ( width=[3,4,6..17/2] )

### 4.3 Exemple de définition d'ensemble de caractéristiques

Voici un exemple de prédicat de caractéristique qui décrit un certain nombre de combinaisons de taille d'image et de niveau de résolution, présument l'enregistrement et l'usage des étiquettes de caractéristique 'Pix-x', 'Pix-y', 'Res-x' et 'Res-y' :

```
( | ( & ( Pix-x=1024)
  ( Pix-y=768)
  ( | ( & ( Res-x=150) ( Res-y=150 )
    ( & ( Res-x=150) ( Res-y=300 )
    ( & ( Res-x=300) ( Res-y=300 )
    ( & ( Res-x=300) ( Res-y=600 )
    ( & ( Res-x=600) ( Res-y=600 ) ) ) )
  ( & ( Pix-x=800)
  ( Pix-y=600)
  ( | ( & ( Res-x=150) ( Res-y=150 )
    ( & ( Res-x=150) ( Res-y=300 )
    ( & ( Res-x=300) ( Res-y=300 )
    ( & ( Res-x=300) ( Res-y=600 )
    ( & ( Res-x=600) ( Res-y=600 ) ) ) ) ;q=0.9
  ( & ( Pix-x=640)
  ( Pix-y=480)
  ( | ( & ( Res-x=150) ( Res-y=150 )
    ( & ( Res-x=150) ( Res-y=300 )
    ( & ( Res-x=300) ( Res-y=300 )
    ( & ( Res-x=300) ( Res-y=600 )
    ( & ( Res-x=600) ( Res-y=600 ) ) ) ) ;q=0.8 )
```

## 5. Correspondance des ensembles de caractéristiques

Cette section présente une procédure pour combiner les ensembles de caractéristiques pour déterminer les collections de caractéristiques communes auxquelles ils se réfèrent, si il en est. Le choix parmi les collections de caractéristiques possibles (fondé sur les q-value ou autrement) n'est pas traité ici.

La confrontation d'un ensemble de caractéristiques à une collection de caractéristiques donnée est par nature très direct : le prédicat de l'ensemble de caractéristiques est simplement évalué par rapport à la collection de caractéristiques donnée, et le résultat (VRAI ou FAUX) indique si la collection de caractéristiques correspond aux capacités, et la valeur de qualité associée peut être utilisée pour le choix entre les diverses collections de caractéristiques.

Confronter un ensemble de caractéristiques à un autre ensemble de caractéristiques est moins direct. Ici, le problème est de déterminer si il y a ou non au moins une collection de caractéristiques qui correspond aux deux ensembles de caractéristiques (par exemple, y a-t-il un chevauchement entre les capacités de caractéristiques d'un format de fichier donné

et les capacités de caractéristiques d'un receveur donné ?)

Cette confrontation d'ensembles de caractéristiques est réalisée par la manipulation logique des expressions de prédicat telles que décrites dans les paragraphes qui suivent.

Pour que cette procédure fonctionne de façon fiable, le prédicat doit être réduit à une forme canonique. La forme canonique utilisée ici est la "forme normale disjonctive". La syntaxe de la forme normale disjonctive est :

```

filtre      = orlist
orlist      = "(" "|" andlist ")" / term
andlist     = "(" "&" termlist ")" / term
termlist    = 1*term
term        = "(" "!" simple ")" / simple

```

où "simple" est comme décrit précédemment au paragraphe 4.1. Donc, la forme canonisée a au moins trois niveaux : un extrême "(...)", une disjonction de "&...", des conjonctions d'essais de valeur de caractéristique éventuellement négatifs.

Note : La forme canonique usuelle pour les expressions de prédicats est "forme clausale". Les procédures pour convertir les expressions générales de prédicat sont données dans [5] (§ 10.2), [11] (§ 2.13) et [12] (§ 5.3.2).

La "forme clausale" pour un prédicat est similaire à "forme normale conjonctive" pour une proposition, étant une conjonction (ET logique) de disjonctions (OU logiques). La forme qui s'y rapporte utilisée ici, qui convient mieux pour les confrontations d'ensembles de caractéristiques, est la "forme normale disjonctive", qui est une disjonction logique (OU) de conjonctions (ET). Sous cette forme, le but de la confrontation des ensembles de caractéristiques est de montrer qu'au moins une des disjonctions peut être satisfaite par une collection de caractéristiques.

Cette prise en considération des formes canoniques est-elle réellement nécessaire ? Après tout, les prédicats de caractéristiques sont juste des expressions booléennes, n'est-ce pas ? Et bien, non : un prédicat de caractéristiques est une expression booléenne qui contient des essais de valeurs de caractéristiques primitives (des comparaisons) représentées par des "item" dans la syntaxe de prédicat de caractéristique. Si ces essais pouvaient tous être supposés indépendamment VRAI ou FAUX, chacun pourrait alors être regardé comme une proposition atomique, et tout le prédicat pourrait être traité conformément à la règle (relativement simple) du calcul propositionnel.

Mais, en général, la même étiquette de caractéristique peut apparaître dans plus d'un "item" d'un prédicat, de sorte que les essais ne peuvent pas être considérés comme indépendants. Bien sûr, l'interdépendance est nécessaire dans toute application significative de confrontation d'ensembles de caractéristiques, et il est important de capturer ces dépendances (par exemple, l'ensemble de résolutions qu'un envoyé peut fournir recoupe t-il l'ensemble de résolutions que peut traiter un receveur ?). Donc, nous avons affaire à des éléments du calcul de prédicats, avec quelques règles supplémentaires pour les manipulations algébriques.

Les descriptions du calcul propositionnel et du calcul de prédicat figurent toutes deux dans [12].

Notre but est de montrer que ces règles supplémentaires sont plus étranges que compliquées. La construction et l'utilisation des prédicats de caractéristiques évite en fait une partie de la complexité du traitement du calcul de prédicat pleinement généralisé.

## 5.1 Stratégie de confrontation d'ensembles de caractéristiques

La stratégie globale pour confronter des ensembles de caractéristiques, est développée ci-dessous :

1. Formuler les hypothèses de confrontation des ensembles de caractéristiques.
2. Remplacer les expressions "set" par les comparaisons équivalentes.
3. Déplacer les négations logiques "vers l'intérieur", de sorte qu'elle s'appliquent toutes directement aux comparaisons de caractéristiques.
4. Éliminer les négations logiques, et exprimer toutes les comparaisons de caractéristiques selon les termes des quatre opérateurs de comparaison.
5. Réduire les hypothèses à la forme canonique disjonctive normale (une disjonction de conjonctions).
6. Pour chaque conjonction, essayer de montrer qu'elle peut être satisfaite par une collection de caractéristiques.
- 6.1. Séparer les essais de valeur de caractéristique en groupes de caractéristiques indépendants, tels que chaque groupe contienne des essais n'impliquant qu'une seule étiquette de caractéristique. Donc, aucun prédicat dans un groupe de caractéristiques ne contient une étiquette de caractéristique qui apparaît aussi dans un autre groupe.
- 6.2. Pour chaque groupe de caractéristiques, fusionner les diverses contraintes en une forme minimale. Ce processus donne soit une expression réduite pour la gamme admissible de valeurs de caractéristiques, soit une expression qui contient la

valeur FAUX, qui indique qu'aucune combinaison de valeurs de caractéristiques ne peut satisfaire les contraintes (auquel cas, la conjonction correspondante ne peut jamais être satisfaite).

7. Si la disjonction restante contient au moins une conjonction satisfaisable, il est alors montré que les contraintes sont satisfaisables.

L'expression finale obtenue par cette procédure, si elle est non vide, peut être utilisée comme déclaration de l'ensemble de caractéristiques résultant pour d'autres opérations de confrontation. C'est à dire qu'elle peut être utilisée comme point de départ pour la combiner avec des prédicats de contrainte supplémentaires d'ensemble de caractéristiques pour déterminer un ensemble de caractéristiques qui soit contraint par les capacités de plusieurs entités dans un chemin de transfert de message.

Note : Comme il est présenté, le processus de confrontation de caractéristiques évalue (et mémorise) toutes les conjonctions de la forme normale disjonctive avant de combiner les comparaisons d'étiquettes de caractéristiques et d'éliminer les conjonctions non satisfaisables. Pour les systèmes à faible mémoire, une autre approche est possible, dans laquelle chaque conjonction de forme normale est énumérée et évaluée tour à tour, celles qui sont satisfaisables étant seules retenues pour la suite du processus.

## 5.2 Formulation du prédicat cible

Une déclaration formelle du problème que nous avons à résoudre peut être donnée de la façon suivante : soit deux prédicats d'ensemble de caractéristiques, "(P x)" et "(Q x)", où 'x' est une collection de caractéristiques, nous souhaitons établir si la proposition suivante est vraie ou non :

$$\text{EXISTS}(x) : (P x) \text{ ET } (Q x)$$

c'est-à-dire existe t-il une collection de caractéristiques 'x' qui satisfasse les deux prédicats, 'P' et 'Q' ?

Puis, si les ensembles de caractéristiques à confronter sont décrits par les prédicats 'P' et 'Q', le problème est de déterminer si il y a un ensemble de caractéristiques qui satisfait le prédicat cible : (& P Q), c'est-à-dire de déterminer si l'ensemble ainsi décrit est non vide.

## 5.3 Remplacer les expressions de "set"

Remplacer toutes les instances de "set" dans le prédicat cible par les formes "simples" équivalentes :

$$\begin{aligned} T = [ E1, E2, \dots En ] & \quad \rightarrow ( (T=[E1]) (T=[E2]) \dots (T=[En]) ) \\ (T=[R1..R2]) & \quad \rightarrow ( \& (T \geq R1) (T \leq R2) ) \\ (T=[E]) & \quad \rightarrow (T=E) \end{aligned}$$

## 5.4 Déplacer vers l'intérieur les négations logiques

Le but de cette étape est de déplacer toutes les négations logiques afin qu'elles s'appliquent directement aux comparaisons de caractéristiques. Durant l'étape suivante, ces négations logiques sont remplacées par des opérateurs de comparaison de remplacement.

Ceci est réalisé par l'application répétée des règles de transformation suivantes :

$$\begin{aligned} (! \& A1 A2 \dots Am ) & \quad \rightarrow ( (! A1 ) (! A2 ) \dots (! Am ) ) \\ (! (! A1 A2 \dots Am ) ) & \quad \rightarrow ( \& (! A1 ) (! A2 ) \dots (! Am ) ) \\ (! (! A ) ) & \quad \rightarrow A \end{aligned}$$

Les deux premières règles sont des formes étendues de la Loi de Morgan, et la troisième est l'élimination des doubles négations.

## 5.5 Remplacer les comparaisons et négations logiques

Les prédicats sont déduits de la syntaxe décrite précédemment, et contiennent les fonctions d'essai de valeur primitive '=',

' $\leq$ ', ' $\geq$ '. Les essais de primitive ont un certain nombre de propriétés bien connues qui sont exploitées pour atteindre une conclusion utile ; par exemple,

$$(A = B) \ \& \ (B = C) \Rightarrow (A = C)$$

$$(A \leq B) \ \& \ (B \leq C) \Rightarrow (A \leq C)$$

Ces règles forment un corps central de déclarations logiques contre lesquelles le prédicat cible peut être évalué. La forme sous laquelle ces déclarations sont exprimées est importante pour réaliser un algorithme de confrontation de prédicats efficace (c'est-à-dire qui ne se met pas en boucle ou échoue à trouver un résultat valide). La première étape de formulation de ces règles est de simplifier le cadre des prédicats primitifs.

Les prédicats primitifs à partir desquels les définitions d'ensembles de caractéristiques sont construites sont '=', ' $\leq$ ' et ' $\geq$ '. On observe que, étant donnée toute paire de valeurs de caractéristiques, la relation entre elles doit être exactement une des suivantes :

(LT a b) : 'a' est inférieur à 'b'.  
 (EQ a b) : 'a' est égal à 'b'.  
 (GT a b) : 'a' est supérieur à 'b'.  
 (NE a b) : 'a' est non égal à 'b', et n'est pas inférieur ou supérieur à 'b'.

(Le cas final survient lorsque sont comparées deux valeurs pour lesquelles aucune relation d'ordre n'est définie, et les valeurs ne sont pas égales ; par exemple, deux valeurs de chaînes inégales.)

Ces quatre cas peuvent être traduits par une paire de prédicats primitifs :

(LE a b) : 'a' est inférieur ou égal à 'b'.  
 (GE a b) : 'a' est supérieur ou égal à 'b'.

Les quatre cas décrits ci-dessus sont pré-présentés par les combinaisons de valeurs de prédicat primitif suivantes :

(LE a b)	(GE a b)	relation
VRAI	FAUX	(LT a b)
VRAI	VRAI	(EQ a b)
FAUX	VRAI	(GT a b)
FAUX	FAUX	(NE a b)

Donc, les trois essais de primitive originaux peuvent être traduits en combinaisons de juste LE et GE, réduisant le nombre de relations supplémentaires qui doivent être capturées ensuite :

$(a \leq b) \quad \rightarrow (LE \ a \ b)$   
 $(a \geq b) \quad \rightarrow (GE \ a \ b)$   
 $(a = b) \quad \rightarrow (\& \ (LE \ a \ b) \ (GE \ a \ b))$

De plus, les négations logiques des trois essais de primitive originaux peuvent être éliminées par l'introduction des primitives 'pas-supérieur' et 'pas-inférieur'.

$(NG \ a \ b) \equiv (! \ (GE \ a \ b))$   
 $(NL \ a \ b) \equiv (! \ (LE \ a \ b))$

en utilisant les règles de transformation suivantes :

$(! \ (a = b)) \quad \rightarrow (! \ (NL \ a \ b) \ (NG \ a \ b))$   
 $(! \ (a \leq b)) \quad \rightarrow (NL \ a \ b)$   
 $(! \ (a \geq b)) \quad \rightarrow (NG \ a \ b)$

Donc, nous avons des règles pour transformer toutes les comparaisons et négations logiques en combinaisons de juste quatre opérateurs relationnels.

## 5.6 Conversion en forme canonique

Note : Les négations logiques ont été éliminées à l'étape précédente.

Développer les disjonctions entre parenthèses, et réduire les conjonctions et disjonctions entre parenthèses :

$$\begin{aligned}
 (& (| A1 A2 \dots Am ) B1 B2 \dots Bn ) \quad \rightarrow \quad (| ( & A1 B1 B2 \dots Bn ) \\
 & \quad \quad \quad ( & A2 B1 B2 \dots Bn ) \\
 & \quad \quad \quad \vdots \\
 & \quad \quad \quad ( & Am B1 B2 \dots Bn ) ) \\
 (& ( & A1 A2 \dots Am ) B1 B2 \dots Bn ) \quad \rightarrow \quad ( & A1 A2 \dots Am B1 B2 \dots Bn ) \\
 (| (| A1 A2 \dots Am ) B1 B2 \dots Bn ) \quad \rightarrow \quad (| A1 A2 \dots Am B1 B2 \dots Bn )
 \end{aligned}$$

Le résultat est en "forme disjonctive normale", une disjonction de conjonctions :

$$\begin{aligned}
 (| ( & S11 S12 \dots ) \\
 & ( & S21 S22 \dots ) \\
 & \quad \quad \quad \vdots \\
 & ( & Sm1 Sm2 \dots Smm ) )
 \end{aligned}$$

où les éléments "Sij" sont des formes de comparaison de caractéristique simple construites durant l'étape du paragraphe 5.5. Chaque terme au sein de la construction "(|...)" de niveau supérieur représente un seul ensemble de caractéristiques possible qui satisfait l'objectif. Noter que l'ordre des entrées au sein des '(|...)' de niveau supérieur, et au sein de chaque '&...', est immatériel.

À partir de là, chaque conjonction '&...' est traitée séparément. Une seule d'entre elles a besoin d'être satisfaisable pour que le but original soit satisfaisable.

(Un livre de conversion en forme clausale [5], [11] utilise des règles légèrement différentes pour donner une "forme conjonctive normale".)

## 5.7 Groupement des prédicats de caractéristiques

Note : On se rappellera qu'à partir de maintenant, chaque conjonction est traitée séparément.

Chaque prédicat de caractéristique simple contient une étiquette de caractéristique "de gauche" et une valeur de caractéristique "de droite" à laquelle elle est comparée.

Pour arranger cela en groupes indépendants, les prédicats simples sont groupés conformément à leur étiquette de caractéristique de gauche ('f').

## 5.8 Fusion des contraintes de caractéristiques seules

Au sein de chaque groupe, appliquer les règles de simplification de prédicat données ci-dessous pour éliminer les contraintes de caractéristique seule redondantes. Tous les prédicats de caractéristique seule sont réduits à une égalité ou à une contrainte de gamme sur cette caractéristique, éventuellement combinée à un certain nombre de déclarations de non égalité.

Si les contraintes sur toute caractéristique sont trouvées en contradiction (c'est-à-dire qu'elles se résolvent en FAUX selon les règles appliquées) la conjonction contenante n'est pas satisfaisable et peut être éliminée. Autrement, la description résultante est une forme minimale de cette conjonction particulière de la définition de l'ensemble de caractéristiques.

### 5.8.1 Règles pour simplifier les valeurs ordonnées

Ces règles sont applicables lorsque il y a une relation d'ordre entre des valeurs 'a' et 'b' données :

(LE f a) (LE f b)	--> (LE f a),	$a \leq b$	autrement, (LE f b),
(LE f a) (GE f b)	--> FAUX,	$a < b$	
(LE f a) (NL f b)	--> FAUX,	$a \leq b$	
(LE f a) (NG f b)	--> (LE f a),	$a < b$ ,	autrement (NG f b)
(GE f a) (GE f b)	--> (GE f a),	$a \geq b$ ,	autrement (GE f b)
(GE f a) (NL f b)	--> (GE f a)	$a > b$ ,	autrement (NL f b)
(GE f a) (NG f b)	--> FAUX,	$a \geq b$	
(NL f a) (NL f b)	--> (NL f a),	$a \geq b$ ,	autrement (NL f b)
(NL f a) (NG f b)	--> FAUX,	$a \geq b$	

(NG f a) (NG f b) --> (NG f a),  $a \leq b$ , autrement (NG f b)

### 5.8.2 Règles pour simplifier les valeurs non ordonnées

Ces règles sont applicables lorsque il n'y a pas de relation d'ordre applicable aux valeurs 'a' et 'b' données :

(LE f a) (LE f b) --> (LE f a),  $a=b$ , autrement FAUX  
 (LE f a) (GE f b) --> (LE f a),  $a=b$ , autrement FAUX,  
 (LE f a) (NL f b) --> (LE f a)  $a!=b$ , autrement FAUX,  
 (LE f a) (NG f b) --> (LE f a),  $a!=b$ , autrement FAUX,  
 (GE f a) (GE f b) --> (GE f a),  $a=b$ , autrement FAUX,  
 (GE f a) (NL f b) --> (GE f a)  $a!=b$ , autrement FAUX,  
 (GE f a) (NG f b) --> (GE f a)  $a!=b$ , autrement FAUX,  
 (NL f a) (NL f b) --> (NL f a),  $a=b$   
 (NL f a) (NG f b) --> (NL f a),  $a=b$   
 (NG f a) (NG f b) --> (NG f a),  $a=b$

## 6. Autres caractéristiques et questions

### 6.1 Prédicats désignés et auxiliaires

Les prédicats désignés et auxiliaires peuvent servir à deux objets :

- (a) rendre plus faciles à écrire et à comprendre les prédicats complexes, et
- (b) fournir une base éventuelle pour la désignation et l'enregistrement des ensembles de caractéristiques.

#### 6.1.1 Définir un prédicat désigné

Une définition de prédicat désigné a la forme suivante :

named-pred = "(" fname \*pname ")" ":" filtre  
 fname = fanion ; nom du prédicat de la caractéristique  
 pname = jeton ; nom du paramètre formel

'fname' est le nom du prédicat.

'pname' est le nom d'un paramètre formel qui peut apparaître dans le corps du prédicat, et qui est remplacé par une valeur fournie lorsque le prédicat est impliqué.

'filtre' est le corps du prédicat. Il peut contenir des références aux paramètres formels, et peut aussi contenir des références à des étiquettes et autres valeurs de caractéristiques définies dans l'environnement dans lequel le prédicat est invoqué. Les références aux paramètres formels peuvent apparaître n'importe où lorsque une étiquette de caractéristique ('fanion') est permis par la syntaxe pour 'filtre'.

Le seul mécanisme spécifique défini par le présent mémoire pour l'introduction d'un prédicat désigné dans une définition d'ensemble de caractéristiques est celui du "prédicat auxiliaire" décrit plus loin. Des protocoles spécifiques de négociation ou d'autres spécifications peuvent définir d'autres mécanismes.

Note : Il a été suggéré de créer un registre des ensembles de caractéristiques ainsi que des valeurs de caractéristiques individuelles. Un tel registre pourrait être utilisé pour introduire des prédicats désignés correspondants à ces ensembles de caractéristiques dans l'environnement d'une assertion de capacité. Il n'entre pas dans le domaine d'application du présent mémoire d'aller plus loin dans cette discussion.

#### 6.1.2 Invoquer des prédicats désignés

En supposant qu'un prédicat désigné a été introduit dans l'environnement d'une autre prédicat, il peut être invoqué par un filtre 'ext-pred' de la forme :

ext-pred = fname \*param  
 param = expr

Le nombre de paramètres doit correspondre à la définition du prédicat désigné qui est invoqué.

### 6.1.3 Prédicats auxiliaires dans un filtre

Un prédicat auxiliaire est rattaché à une définition de filtre par l'extension suivante à la syntaxe de "filtre" :

```
filtre      =/ "(" filtercomp *( ";" paramètre ) ")" "où" 1*( named-pred ) "fin"
```

Les prédicats désignés introduits par "named-pred" sont visibles à partir du corps du "filtercomp" du filtre auquel ils sont rattachés, mais ne sont pas visibles les uns des autres. Ils ont tous accès au même environnement que "filtre", en plus de leurs propres paramètres formels. (Les règles de portée normales s'appliquent : un paramètre formel avec le même nom qu'une valeur dans l'environnement de "filtre" cache effectivement la valeur de l'environnement au corps du prédicat auquel il s'applique.)

Note : Les prédicats récurrents ne sont pas permis. Les règles de portée devraient le garantir.

### 6.1.4 Correspondance d'une caractéristique avec des prédicats désignés

Les procédures qui précèdent peuvent être étendues pour traiter les prédicats désignés simplement en les instanciant (c'est-à-dire en substituant) les prédicats partout où ils sont invoqués, avant d'effectuer la conversion en forme disjonctive normale. En l'absence de prédicats récurrents, la fin de cette procédure est garantie.

Lorsque on substitue le corps d'un prédicat au moment de son invocation, les instances de paramètres formels au sein du corps du prédicat doivent être remplacés par le paramètre réel correspondant au moment de l'invocation.

### 6.1.5 Exemple

Cet exemple reprend ce qui est donné au paragraphe 4.3 en utilisant un prédicat auxiliaire nommé 'Res' :

```
(| (& (Pix-x=1024) (Pix-y=768) (Res Res-x Res-y) )
  (& (Pix-x=800) (Pix-y=600) (Res Res-x Res-y) );q=0,9
  (& (Pix-x=640) (Pix-y=480) (Res Res-x Res-y) );q=0,8 )
où
(Res Res-x Res-y) :-
  (| (& (Res-x=150) (Res-y=150) )
    (& (Res-x=150) (Res-y=300) )
    (& (Res-x=300) (Res-y=300) )
    (& (Res-x=300) (Res-y=600) )
    (& (Res-x=600) (Res-y=600) ) )
fin
```

Noter que les paramètres formels de "Res", "Res-x" et "Res-y" empêchent le corps du prédicat désigné de faire référence à des valeurs de caractéristiques de dénomination similaire.

## 6.2 Désignation des unités

Dans certains cas exceptionnels, il peut y avoir des conventions différentes pour les unités de mesure d'une caractéristique donnée. Par exemple, la résolution est couramment exprimée en points par pouce (dpi, *dots per inch*) ou en points par centimètre (dpcm, *dots per centimetre*) dans différentes applications (par exemple, impression ou télécopie).

Dans de tels cas, une désignation d'unité peut être ajoutée à une valeur de caractéristique conformément aux conventions indiquées ci-dessous (voir aussi [3]). Ces considérations ne s'appliquent qu'aux caractéristiques qui ont des valeurs numériques.

Toute étiquette de caractéristique a une unité de mesure standard. Toute expression d'une valeur de caractéristique qui utilise cette unité est donnée sans désignation d'unité – c'est le cas normal. Lorsque la valeur de caractéristique est exprimée dans une autre unité, une désignation d'unité est ajoutée à la valeur numérique de la caractéristique.

L'enregistrement d'une étiquette de caractéristique indique l'unité de mesure standard de cette caractéristique, ainsi que

toutes les unités de remplacement et les désignations d'unités correspondantes qui pourraient être utilisées conformément à la RFC2506 [3].

Donc, si l'unité de mesure standard pour la résolution est "dpcm", le prédicat de caractéristique "(res=200)" sera utilisé pour indiquer une résolution de 200 points par centimètre et "(res=72dpi)" pourrait être utilisé pour indiquer 72 points par pouce.

Les désignations d'unités s'accommodent de l'extension suivante à la syntaxe de prédicat de caractéristique :

fvalue        =/ nombre \*WSP jeton

Lorsque on effectue une confrontation d'ensembles de caractéristiques, les comparaisons de caractéristiques avec et sans désignations d'unités, ou les comparaisons avec des désignations d'unités différentes, sont traitées comme si elles étaient des caractéristiques différentes. Donc, le prédicat de caractéristique "(res=200)" ne devrait, en général, pas échouer à correspondre au prédicat "(res=200dpi)".

Note : Un processeur de protocole ayant une connaissance spécifique de la caractéristique et des unités concernées peut reconnaître les relations entre les prédicats de caractéristiques dans l'exemple ci-dessus, et échouer à faire correspondre ces prédicats. Cela paraît être un comportement naturel dans cet exemple simple, mais peut causer une complexité supplémentaire dans des cas plus généraux. En conséquence, ceci n'est pas considéré comme exigé ou comme un comportement normal. On suppose qu'une application concernée va s'assurer de la cohérence du traitement des caractéristiques en adoptant une unité cohérente pour une caractéristique donnée.

### 6.3 Types inconnus de données de valeur de caractéristique

Le présent mémoire a traité des valeurs de caractéristiques qui ont des propriétés de comparaisons bien comprises : nombres avec des relations d'égalité, de plus grand ou plus petite que, et autres valeurs avec des relations d'égalité seulement.

Certaines valeurs de caractéristiques peuvent avoir des opérations de comparaison qui ne sont pas couvertes par ce cadre. Par exemple, des chaînes contenant des numéros de version multi-parties : "x.y.z". De telles comparaisons de caractéristiques ne sont pas couvertes par le présent mémoire.

Des applications spécifiques peuvent reconnaître et traiter des étiquettes de caractéristiques qui sont associées à de telles valeurs. Des travaux futurs pourront définir des moyens d'introduire de nouveaux types de données de valeur de caractéristique qui permettent de les utiliser dans des applications qui ne contiennent pas l'incorporation de la connaissance de leurs propriétés.

## 7. Exemples et commentaires supplémentaires

### 7.1 Exemple élaboré

Cet exemple considère l'envoi d'un document à un système de télécopie en noir et blanc qui a les capacités de réception suivantes :

```
(& (dpi=[200,300])
  (grey=2) (color=0)
  (image-coding=[MH,MR]) )
```

En ce qui concerne le document lui-même, on suppose qu'il est disponible chez l'envoyeur sous trois formats possibles, A4 à haute résolution, B4 basse résolution et A4 couleur haute résolution, décrits par :

```
(& (dpi=300)
  (grey=2)
  (image-coding=MR) )
```

```
(& (dpi=200)
  (grey=2)
  (image-coding=[MH,MMR]) )
```

```
(& (dpi=300) (dpi-xratio=1)
  (color<=256)
  (image-coding=JPEG) )
```

Ces trois formats d'image peuvent être combinés en une déclaration de capacité composite par une opération OU logique (pour décrire format-1 OU format-2 OU format-3):

```
(| (& (dpi=300)
  (grey=2)
  (image-coding=MR) )
  (& (dpi=200)
  (grey=2)
  (image-coding=[MH,MMR])) )
  (& (dpi=300)
  (color<=256)
  (image-coding=JPEG) ) )
```

La description de document composite peut être confrontée à la description des capacités du receveur en combinant les descriptions de capacités avec l'opération logique ET :

```
(& (& (dpi=[200,300])
  (grey=2) (color=0)
  (image-coding=[MH,MR]))
  (| (& (dpi=300)
  (grey=2)
  (image-coding=MR) )
  (& (dpi=200)
  (grey=2)
  (image-coding=[MH,MMR])) )
  (& (dpi=300)
  (color<=256)
  (image-coding=JPEG) ) ) )
```

--> Notation d'ensemble de valeurs étendue :

```
(& (& (| (dpi=200) (dpi=300) )
  (grey=2) (color=0)
  (| (image-coding=MH) (image-coding=MR) ) )
  (| (& (dpi=300)
  (grey=2)
  (image-coding=MR) )
  (& (dpi=200)
  (grey=2)
  (| (image-coding=MH) (image-coding=MMR) ) )
  (& (dpi=300)
  (color<=256)
  (image-coding=JPEG) ) ) )
```

--> En réduisant les '&...' incorporés :

```
(& (| (dpi=200) (dpi=300) )
  (grey=2) (color=0)
  (| (image-coding=MH) (image-coding=MR) )
  (| (& (dpi=300)
  (grey=2)
  (image-coding=MR) )
  (& (dpi=200)
  (grey=2)
  (| (image-coding=MH) (image-coding=MMR) ) )
  (& (dpi=300)
  (color<=256)
  (image-coding=JPEG) ) ) )
```

--> (en répartissant les '&...)' sur les '(!...)' internes :

```
(& (! (dpi=200) (dpi=300) )
  (grey=2) (color=0)
  (! (image-coding=MH) (image-coding=MR) )
  (! (& (dpi=300) (grey=2) (image-coding=MR) )
    (& (dpi=200) (grey=2) (image-coding=MH) )
    (& (dpi=200) (grey=2) (image-coding=MMR) )
    (& (dpi=300) (color<=256) (image-coding=JPEG) ) ) )
```

--> continuer à répartir les '&...)' sur '(!...)', et réduire les '&...)' et '(!...)' incorporés ... :

```
(! (& (dpi=200) (grey=2) (color=0) (image-coding=MH)
  (! (& (dpi=300) (grey=2) (image-coding=MR) )
    (& (dpi=200) (grey=2) (image-coding=MH) )
    (& (dpi=200) (grey=2) (image-coding=MMR) )
    (& (dpi=300) (color<=256) (image-coding=JPEG) ) ) )
(& (dpi=200) (grey=2) (color=0) (image-coding=MR)
  (! (& (dpi=300) (grey=2) (image-coding=MR) )
    (& (dpi=200) (grey=2) (image-coding=MH) )
    (& (dpi=200) (grey=2) (image-coding=MMR) )
    (& (dpi=300) (color<=256) (image-coding=JPEG) ) ) )
(& (dpi=300) (grey=2) (color=0) (image-coding=MH)
  (! (& (dpi=300) (grey=2) (image-coding=MR) )
    (& (dpi=200) (grey=2) (image-coding=MH) )
    (& (dpi=200) (grey=2) (image-coding=MMR) )
    (& (dpi=300) (color<=256) (image-coding=JPEG) ) ) )
(& (dpi=300) (grey=2) (color=0) (image-coding=MR)
  (! (& (dpi=300) (grey=2) (image-coding=MR) )
    (& (dpi=200) (grey=2) (image-coding=MH) )
    (& (dpi=200) (grey=2) (image-coding=MMR) )
    (& (dpi=300) (color<=256) (image-coding=JPEG) ) ) )
```

--> ... jusqu'à réaliser la forme normale :

```
(! (& (dpi=200) (grey=2) (color=0) (image-coding=MH)
  (dpi=300) (grey=2) (image-coding=MR) )
  (& (dpi=200) (grey=2) (color=0) (image-coding=MR)
  (dpi=300) (grey=2) (image-coding=MR) )
  (& (dpi=300) (grey=2) (color=0) (image-coding=MH)
  (dpi=300) (grey=2) (image-coding=MR) )
  (& (dpi=300) (grey=2) (color=0) (image-coding=MR)
  (dpi=300) (grey=2) (image-coding=MR) )
  (& (dpi=200) (grey=2) (color=0) (image-coding=MH)
  (dpi=200) (grey=2) (image-coding=MH) )
  (& (dpi=200) (grey=2) (color=0) (image-coding=MR)
  (dpi=200) (grey=2) (image-coding=MH) )
  (& (dpi=300) (grey=2) (color=0) (image-coding=MH)
  (dpi=200) (grey=2) (image-coding=MH) )
  (& (dpi=300) (grey=2) (color=0) (image-coding=MR)
  (dpi=200) (grey=2) (image-coding=MH) )
  (& (dpi=200) (grey=2) (color=0) (image-coding=MH)
  (dpi=200) (grey=2) (image-coding=MMR) )
  (& (dpi=200) (grey=2) (color=0) (image-coding=MR)
  (dpi=200) (grey=2) (image-coding=MMR) )
  (& (dpi=300) (grey=2) (color=0) (image-coding=MH)
  (dpi=200) (grey=2) (image-coding=MMR) )
  (& (dpi=300) (grey=2) (color=0) (image-coding=MR)
  (dpi=200) (grey=2) (image-coding=MMR) )
  (& (dpi=200) (grey=2) (color=0) (image-coding=MH)
  (dpi=300) (color<=256) (image-coding=JPEG) ) )
  (& (dpi=200) (grey=2) (color=0) (image-coding=MR)
  (dpi=300) (color<=256) (image-coding=JPEG) ) )
```

```
(& (dpi=300) (grey=2) (color=0) (image-coding=MH)
  (dpi=300) (color<=256) (image-coding=JPEG) ) )
(& (dpi=300) (grey=2) (color=0) (image-coding=MR)
  (dpi=300) (color<=256) (image-coding=JPEG) ) )
```

--> Grouper les termes dans chaque conjonction par étiquette de caractéristique :

```
(| (& (dpi=200) (dpi=300) (grey=2) (grey=2) (color=0)
  (image-coding=MH) (image-coding=MR) )
  (& (dpi=200) (dpi=300) (grey=2) (grey=2) (color=0)
  (image-coding=MR) (image-coding=MR) )
  :
  (etc.)
  :
  (& (dpi=300) (dpi=300) (grey=2) (color=0) (color<=256)
  (image-coding=MR) (image-coding=JPEG) ) )
```

--> Combiner les comparaisons d'étiquettes et éliminer les conjonctions non satisfaisables :

```
(| (& (dpi=300) (grey=2) (color=0) (image-coding=MR) )
  (& (dpi=200) (grey=2) (color=0) (image-coding=MH) ) )
```

Donc, on voit que cette combinaison d'options d'envoyeur et de receveur peut transférer une image à deux niveaux, soit à 300 dpi en utilisant le codage MR, soit à 200 dpi en utilisant le codage MH.

Points à noter sur le processus de confrontation des caractéristiques :

- o L'option de document couleur est éliminée parce que le receveur ne peut traiter ni la couleur (indiqué par '(color=0)') ni le codage JPEG.
- o La version haute résolution du document avec '(dpi=300)' doit être envoyée en utilisant '(image-coding=MR)' parce que c'est le seul codage des données d'image disponible que le revceveur peut utiliser pour les documents en haute définition. (Les codages disponibles à 300dpi du document sont ici MMR et MH, et les capacités du receveur sont MH et MR.)

## 7.2 Note sur la portée d'une étiquette de caractéristique

Ce paragraphe contient des commentaires complémentaires sur l'interprétation des prédicats d'ensembles de caractéristiques. Il n'étend pas ni ne modifie ce qui a été décrit précédemment. Il essaye plutôt de préciser de possibles malentendus.

Le fait essentiel qui doit être établi ici est que : au sein d'une collection de caractéristiques données, chaque étiquette de caractéristique ne peut avoir qu'une seule valeur.

L'idée est expliquée ci-dessous dans le contexte de l'utilisation d'un cadre de caractéristiques de support pour décrire les caractéristiques des données d'image transmises.

Dans ce contexte, nous avons l'exigence que toute valeur d'étiquette de caractéristique doit s'appliquer à l'image entière, et ne peut pas avoir des valeurs différentes pour les différentes parties d'une image. Ceci est une conséquence de la façon dont le cadre des prédicats de caractéristiques est utilisé pour décrire les différentes images possibles, comme les différentes images qui peuvent être restituées par un receveur donné.

Cette idée est illustrée ici en utilisant un exemple d'une description fautive d'un ensemble de caractéristiques fondée sur le format d'image TIFF défini pour l'utilisation de la télécopie Internet [13] :

```
(& (& (MRC-mode=1) (stripe-size=256) )
  (| (& (image-coding=JBIG-2-LEVEL) (stripe-size=128) )
  (image-coding=[MH,MR,MMR]) ) )
```

Cet exemple est révélateur parce que l'attribut "stripe-size" est appliqué de façon différente aux différents attributs sur des données formatées en MRC : il peut être appliqué au format MRC comme un tout, et il peut être appliqué séparément à une

image JBIG qui apparaîtrait comme une partie des données MRC.

On peut imaginer que cet exemple décrit une taille de bande de 256 lorsque appliquée au format d'image MRC, et une taille de bande séparée de 128 lorsque appliquée à une image codée en JBIG-2-LEVEL au sein des données formatées en MRC. Mais cela ne fonctionne pas comme cela : les prédicats utilisés obéissent aux lois normales de la logique booléenne, et seront transformés comme suit :

--> [Réduire les (&...)] incorporés :

```
(& (MRC-mode=1) (stripe-size=256)
  (| (& (image-coding=JBIG-2-LEVEL) (stripe-size=128) )
    (image-coding=[MH,MR,MMR]) ) )
```

--> [Répartir (&...) sur (...)] :

```
(| (& (MRC-mode=1) (stripe-size=256)
    (& (image-coding=JBIG-2-LEVEL) (stripe-size=128) ) )
  (& (MRC-mode=1) (stripe-size=[0..256])
    (image-coding=[MH,MR,MMR]) ) )
```

--> [Réduire les (&...) incorporés et les étiquettes de groupes de caractéristiques] :

```
(| (& (MRC-mode=1)
    (stripe-size=256)
    (stripe-size=128)
    (image-coding=JBIG-2-LEVEL) )
  (& (MRC-mode=1)
    (stripe-size=256)
    (image-coding=[MH,MR,MMR]) ) )
```

L'examen de cette expression finale montre qu'on exige à la fois " stripe-size=128" et "stripe-size=256" au sein de la même conjonction. Ceci est manifestement faux, de sorte que la conjonction entière doit être fausse, ce qui réduit l'expression entière du prédicat à :

```
(& (MRC-mode=1)
  (stripe-size=256)
  (image-coding=[MH,MR,MMR]) ) )
```

Ceci indique qu'aucunes données formatées en MRC contenant une image codée en JBIG-2-LEVEL ne sont permises au sein de l'ensemble de caractéristiques, ce qui n'est pas ce qui était voulu dans ce cas.

La seule façon d'éviter cela dans des situations où une caractéristique donnée a des contraintes différentes dans les différentes parties d'une ressource est d'utiliser des étiquettes de caractéristique distinctes. Dans cet exemple, "MRC-stripe-size" et "JBIG-stripe-size" pourraient être utilisés pour rendre cette intention :

```
(& (& (MRC-mode=1) (MRC-stripe-size=256) )
  (| (& (image-coding=JBIG-2-LEVEL) (JBIG-stripe-size=128) )
    (image-coding=[MH,MR,MMR]) ) )
```

qui va se réduire à :

```
(| (& (MRC-mode=1)
    (MRC-stripe-size=256)
    (JBIG-stripe-size=128)
    (image-coding=JBIG-2-LEVEL) )
  (& (MRC-mode=1)
    (MRC-stripe-size=256)
    (image-coding=[MH,MR,MMR]) ) )
```

La propriété du cadre de description de capacité expliquée ci-dessus est rendue par l'idée d'une "collection de caractéristiques" qui (dans ce contexte) décrit les valeurs de caractéristiques qui s'appliquent à une seule ressource. Au sein d'une collection de caractéristiques, chaque étiquette de caractéristique ne peut avoir plus d'une valeur.

Les caractéristiques d'un expéditeur ou receveur d'image sont décrites par un "Ensemble de caractéristiques", qui est formellement un ensemble de collections de caractéristiques. Ici, le prédicat de l'ensemble de caractéristiques est appliqué à une collection de caractéristiques d'image pour déterminer si il appartient ou non à l'ensemble qui peut être traité par un

receveur de l'image.

## 8. Considérations pour la sécurité

Certains problèmes de sécurité concernant la négociation du contenu sont soulevés dans [1], [2], [3].

Les problèmes principaux pour la sécurité dans les mécanismes d'identification des capacités sont :

- o La divulgation involontaire d'informations privées à travers l'annonce de capacités ou de préférences de l'utilisateur.
- o L'interruption du fonctionnement du système causée par la fourniture accidentelle ou malveillante d'informations de capacités incorrectes.
- o L'utilisation d'un mécanisme d'identification de capacités qui pourrait être utilisé pour sonder un réseau (par exemple en identifiant les hôtes spécifiques utilisés, et en exploitant leurs faiblesses connues).

Les problèmes de sécurité les plus préoccupants sont soulevés par les mécanismes qui envoient automatiquement les données d'identification de capacités en réponse à une interrogation d'un système inconnu. L'utilisation de services de répertoire (fondés sur LDAP [7], etc.) semble poser moins de problèmes parce que des mécanismes d'authentification appropriés sont disponibles.

Les mécanismes qui fournissent les informations sur les capacités lors de l'envoi d'un message posent moins de problèmes, vraisemblablement parce que on peut estimer qu'une personne qui divulgue ces détails a l'intention de communiquer avec le receveur de ces détails. Cela ne résout cependant pas les problèmes de fourniture déguisée d'informations de capacité incorrectes.

L'utilisation de passerelles de conversion de format peut se révéler problématique parce que de tels systèmes tendraient à vaincre tous les mécanismes de vérification d'intégrité et d'authenticité de message employés.

## 9. Remerciements

Des remerciements sont dus à Larry Masinter pour sa démonstration de l'ampleur de la question des caractéristiques des supports, et pour ses encouragements au développement des premières réflexions sur le sujet.

Baucoup des idées présentées découlent du travail intitulé "Négociation transparente du contenu dans HTTP" de Koen Holtman et Andy Mutz [4].

Les premières discussions sur les idées au sein des groupes de travail HTTP et FAX de l'IETF ont conduit à d'autres apports utiles de la part de Koen Holtman, Ted Hardie et Dan Wing. Le débat s'est ensuite déplacé dans le groupe de travail "conneg" de l'IETF, où Al Gilman et Koen Holtman ont été d'une aide particulière en précisant l'algèbre d'ensemble de caractéristique. Les idées de préférences et d'unités spécifiques ont été suggérées par Larry Masinter.

Le présent travail a bénéficié du soutien de Content Technologies Ltd et 5th Generation Messaging Ltd.

## 10. Références

- [1] Hardie, T., "Scenarios for the Delivery of Negotiated Content", Travail en cours.
- [2] G. Klyne, "Cadre de négociation de contenu indépendant du protocole", RFC2703, septembre 1999. (*Information*)
- [3] K. Holtman, A. Mutz, T. Hardie, "Procédure d'enregistrement d'[étiquette de caractéristique](#) de support", RFC2506, mars 1999. ([BCP0031](#))
- [4] K. Holtman, A. Mutz, "Négociation de contenu transparent dans HTTP", RFC2295, mars 1998. (*Expérimentale*)
- [5] W. F. Clocksin et S. Mellish, "Programming in Prolog" (2<sup>me</sup> édition), Springer Verlag, ISBN 3-540-15011-0 / 0-387-15011-0, 1984.
- [6] L. Masinter et autres, "Caractéristiques de support pour l'affichage, l'impression et la télécopie", RFC2534, mars 1999. (*P.S.*)

- [7] M. Wahl, T. Howes et S. Kille, "Protocole léger d'[accès à un répertoire](#) (v3)", RFC2251, décembre 1997.
- [8] T. Howes, "Représentation comme chaîne des [filtres de recherche](#) LDAP", RFC2254, décembre 1997. (*Obsolète, voir [RFC4510](#), [RFC4515](#)*) (P.S.)
- [9] R. Fielding, J. Gettys, J. Mogul, H. Frystyk, T. Berners-Lee, "Protocole de transfert Hypertext -- [HTTP/1.1](#)", RFC2068, janvier 1997. (*Obsolète, voir [RFC2616](#)*) (P.S.)
- [10] D. Crocker et P. Overell, "BNF augmenté pour les spécifications de syntaxe : [ABNF](#)", RFC2234, novembre 1997. (*Obsolète, voir [RFC5234](#)*)
- [11] Peter Gray, "Logic, Algebra and Databases", Ellis Horwood Series : Computers and their Applications, ISBN 0-85312-709-3/0-85312-803-3 (Ellis Horwood Ltd), ISBN 0-470-20103-7/0-470-20259-9 (Halstead Press), 1984.
- [12] Edmund Burk et Eric Foxley, "Logic and its Applications", Prentice Hall, Series in computer science, ISBN 0-13-030263-5, 1996.
- [13] L. McIntyre et autres, "Format de fichier pour la télécopie Internet", RFC2301, mars 1998. (*Obsolète, voir [RFC3949](#)*) (P.S.)
- [14] Algorithme Apache de négociation de contenu : <<http://www.apache.org/docs/content-negotiation.html>>

## 11. Adresse de l'auteur

Graham Klyne	
Content Technologies Ltd.	5th Generation Messaging Ltd.
Forum 1	5 Watlington Street
Station Road	Nettlebed
Theale	Henley-on-Thames
Reading, RG7 4RA	RG9 5AB
United Kingdom	United Kingdom.
téléphone : +44 118 930 1300	+44 1491 641 641
télécopie : +44 118 930 1301	+44 1491 641 611
mél : <a href="mailto:GK@ACM.ORG">GK@ACM.ORG</a>	

## Déclaration complète des droits de reproduction

Copyright (C) The Internet Society (1999). Tous droits réservés.

Le présent document et ses traductions peuvent être copiés et fournis aux tiers, et les travaux dérivés qui les commentent ou les expliquent ou aident à leur mise en œuvre peuvent être préparés, copiés, publiés et distribués, en tout ou partie, sans restriction d'aucune sorte, pourvu que la déclaration de copyright ci-dessus et le présent paragraphe soient inclus dans toutes copies et travaux dérivés. Cependant, le présent document lui-même ne peut être modifié d'aucune façon, en particulier en retirant la notice de droits de reproduction ou les références à la Internet Society ou aux autres organisations Internet, excepté autant qu'il est nécessaire pour les besoins du développement des normes Internet, auquel cas les procédures de droits de reproduction définies dans les procédures des normes Internet doivent être suivies, ou pour les besoins de la traduction dans d'autres langues que l'anglais.

Les permissions limitées accordées ci-dessus sont perpétuelles et ne seront pas révoquées par la Internet Society ou ses successeurs ou ayant droits.

Le présent document et les informations y contenues sont fournies sur une base "EN L'ÉTAT" et le contributeur, l'organisation qu'il ou elle représente ou qui le/la finance (s'il en est), la INTERNET SOCIETY et la INTERNET ENGINEERING TASK FORCE déclinent toutes garanties, exprimées ou implicites, y compris mais non limitées à toute garantie que l'utilisation des informations ci encloses ne violent aucun droit ou aucune garantie implicite de commercialisation ou d'aptitude à un objet particulier.

## Remerciement

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