

## NORME ST.96 – ANNEXE I

### REGLES ET CONVENTIONS DE CONCEPTION XML

Version 5.0

*Révision approuvée par l'Équipe d'experts de la norme XML4IP  
du Comité des normes de l'OMPI (CWS) le 1<sup>er</sup> octobre 2021*

#### Table des matières

NORME ST.96 – ANNEXE I .....	1
1. INTRODUCTION .....	3
1.1 Aperçu.....	3
1.2 Portée .....	3
1.3 Comment utiliser le présent document.....	3
1.4 Structure du document .....	3
1.5 Terminologie et notation .....	4
1.5.1 Mots clés.....	4
1.5.2 Notations générales.....	4
1.5.3 Identificateurs de règle .....	4
2. CONVENTIONS DE CONCEPTION XML.....	5
2.1 Règles de conception générales XML.....	5
2.2 Conventions de nommage XML .....	5
2.2.1 Conventions de nommage de construction de schémas .....	5
2.2.2 Conventions de nommage des fichiers d'un schéma.....	7
2.3 Stratégie de modularité .....	7
2.3.1 Modules d'un schéma.....	7
2.3.2 Renvoi à des schémas externes .....	8
2.4 Réutilisabilité.....	9
2.5 Espaces de nommage.....	9
2.5.1 Déclaration et qualification des espaces de nommage.....	9
2.5.2 Espaces de nommage dans le schéma XML.....	10
2.5.3 Espaces de nommage cibles.....	10
2.5.4 Espaces de nommage implicites.....	12
2.6 Versions de schéma .....	12
2.6.1 Modifications majeures et mineures.....	12
2.6.1.1 Versions majeures.....	12
2.6.1.2 Versions mineures.....	12
2.6.2 Stratégie de contrôle de version du schéma .....	13
2.6.2.1 Dossier et conventions de nommage des fichiers dans le contrôle de version du schéma .....	13
2.6.2.2 Attribut <code>version</code> du schéma XML incorporé dans le contrôle de version du schéma .....	13
2.6.2.3 Attributs <code>st96Version</code> et <code>ipoVersion</code> définis par l'utilisateur dans le contrôle de version du schéma pour les instances XML.....	14
2.7 Possibilités de conversion avec d'autres normes XML de l'OMPI .....	16
2.8 Schémas de normes industrielles.....	17
3. CONVENTIONS DE CONSTRUCTION DE SCHÉMAS XML.....	17
3.1 Définitions des types.....	17
3.1.1 Types simples.....	17
3.1.1.1 Types de données incorporées du W3C.....	17

3.1.1.2	Types de données définies par l'utilisateur .....	18
3.1.2	Types complexes.....	18
3.2	Éléments et attributs.....	18
3.2.1	Éléments ou attributs.....	18
3.2.2	Éléments.....	19
3.2.2.1	Cardinalité des éléments.....	19
3.2.2.2	Éléments vides.....	19
3.2.3	Attributs.....	19
3.2.4	Regroupement des éléments et attributs .....	19
3.3	Extension et restriction .....	20
3.3.1	Extension.....	20
3.3.2	Restriction.....	20
3.3.3	Groupes de substitution.....	20
3.4	Contraintes en matière d'identité.....	20
3.5	Documentation du schéma.....	21
3.5.1	Documentation sur la structure du schéma.....	21
3.5.2	Documentation relative à l'en-tête du schéma .....	21
4.	RÈGLES DE CONCEPTION DES INSTANCES.....	22
4.1	Espaces de nommage dans les documents d'instance XML.....	22
4.1.1	Validation des documents d'instance XML.....	22
4.1.2	Déclaration et qualification de l'espace de nommage dans les documents d'instance XML.....	23
4.1.3	L'espace de nommage de l'instance du schéma W3C .....	24
4.1.4	Portée de l'espace de nommage.....	24
4.2	Entités externes .....	24
	<b>APPENDICE A – LISTE DES RÈGLES DE CONCEPTION.....</b>	<b>25</b>
	Règles de conception générales .....	25
	Règles de conception d'un schéma.....	26
	Règles de conception de l'instance .....	29
	<b>APPENDICE B – TERMES DE REPRÉSENTATION.....</b>	<b>31</b>
	<b>APPENDICE C – LISTE DES SIGLES ET ABRÉVIATIONS .....</b>	<b>32</b>
	<b>APPENDICE D – RÉFÉRENCES.....</b>	<b>35</b>
	Normes de l'OMPI .....	35
	Normes industrielles .....	35

## 1. INTRODUCTION

### 1.1 *Aperçu*

1. Les règles et conventions de conception XML donnent l'occasion de promouvoir l'harmonie entre tous les types de titres de propriété intellectuelle définis dans la norme ST.96 de l'OMPI et de faciliter l'échange de données entre les offices de propriété intellectuelle en fournissant des schémas XML réutilisables et interopérables.

### 1.2 *Portée*

2. Le présent document vise à établir une série complète de règles et de conventions pour la création et l'utilisation des schémas et instances XML couvrant tous les types d'information en matière de propriété intellectuelle afin de faciliter le dépôt, le traitement, la publication et l'échange de données entre les spécialistes de la propriété intellectuelle.

### 1.3 *Comment utiliser le présent document*

3. Le présent document est destiné à être utilisé par les offices de propriété intellectuelle, les fournisseurs de données de propriété intellectuelle et l'ensemble de la communauté de la propriété intellectuelle. Cette dernière devrait utiliser le présent document comme point de référence concernant les règles et conventions de conception approuvées que tous les schémas XML doivent suivre pour être considérés comme compatibles avec la norme ST.96. Les offices de propriété intellectuelle peuvent s'inspirer de ce document pour élaborer leurs propres règles de conception internes. Il convient de consulter le présent document avant de commencer à élaborer un nouveau schéma XML ou de modifier un schéma XML existant. Une fois le schéma XML élaboré, le présent document devrait être utilisé pour vérifier sa conformité avec les règles de conception.

### 1.4 *Structure du document*

4. Il est recommandé de lire le présent document dans l'ordre dans lequel il a été rédigé. Ce guide est structuré comme suit :

- La Section 1, Introduction, décrit les règles générales qui s'appliquent dans tout le document;
- La Section 2, Conventions de conception XML, définit les règles de haut niveau qui s'appliquent aux activités d'élaboration de schémas et d'instances;
- La Section 3, Conventions de construction de schémas XML, définit des règles de conception spécifiques pour l'utilisation des spécifications du schéma W3C pour la création de schémas XML; et
- La Section 4, Règles de conception des instances, définit des règles de conception spécifiques pour la création d'instances.

5. En outre, ce guide contient quatre appendices :

- L'appendice A, Résumé des règles de conception, résume les règles de conception figurant dans le présent document;
- L'appendice B, Termes de représentation, contient la définition des termes de représentation;
- L'appendice C, Liste des sigles et abréviations, contient les termes et abréviations utilisés dans le présent document; et
- L'appendice D, Références, contient des renvois à des normes de l'OMPI et à d'autres normes industrielles.

## 1.5 Terminologie et notation

### 1.5.1 Mots clés

#### 6. En règle générale,

- le terme "schéma XML" désigne un langage servant à décrire la structure des documents XML et à en limiter le contenu; et
- le terme "Schéma XML du Consortium W3C" désigne les schémas XML qui sont pleinement conformes à la série de recommandations sur le langage de définition du schéma XML du Consortium W3C — Schéma XML, Partie 1 : Structures et schémas XML; Partie 2 : Type de données.

#### 7. Dans le présent document,

- le terme schéma désigne le schéma XML défini à l'annexe III de la norme ST.96 de l'OMPI; et
- la composante désigne un type, un élément ou un attribut.

8. Les mots clés DOIT ou DOIVENT, NE DOIT PAS ou NE DOIVENT PAS, DEVRAIT ou DEVRAIENT, NE DEVRAIT PAS ou NE DEVRAIENT PAS, PEUT ou PEUVENT qui figurent dans le présent document doivent être entendus au sens décrit dans la section "Définitions et terminologie" de la norme ST.96 de l'OMPI. Les formes en minuscules de ces mots sont utilisées au sens anglais courant.

### 1.5.2 Notations générales

#### 9. Les notations ci-après sont utilisées d'un bout à l'autre du présent document :

- `<>` : indique un terme utilisé pour décrire un espace qualifié qui, dans l'application, sera remplacé par une valeur d'instance spécifique.
- " " : indique que le texte entre guillemets doit être utilisé *in extenso* dans l'application.
- { } : indique que l'application est facultative.
- Caractères `Courier` : indique les mots clés XML, les noms de balise et les codes XML.

### 1.5.3 Identificateurs de règle

#### 10. Toutes les règles de conception sont normatives. Les règles de conception sont identifiées au moyen d'un préfixe de [XX-*nn*].

##### a) La valeur "XX" est un préfixe qui sert à classer les types de règle comme suit :

- GD pour les règles de conception générales;
- SD pour les règles de conception du schéma; et
- ID pour les règles de conception des instances.

b) La valeur "*nn*" définit le prochain numéro disponible dans l'ordre d'un type de règle donné. Il y a lieu de noter que le numéro ne désigne pas la position de la règle, en particulier pour une nouvelle règle. Une nouvelle règle sera placée dans le contexte pertinent. Par exemple, l'identificateur [GD-40] identifie la quarantième règle de conception générale. La règle [GD-40] peut être placée entre les règles [GD-20] et [GD-21] au lieu de suivre [GD-39] s'il s'agit de la position la plus appropriée pour cette règle.

c) L'identificateur de la règle supprimée sera conservé tandis que la règle sera remplacée par le terme "Supprimé".

## 2. CONVENTIONS DE CONCEPTION XML

### 2.1 Règles de conception générales XML

11. La présente section du guide contient des règles et principes de conception XML généraux et de haut niveau qui s'appliquent à toutes les activités de développement XML plutôt qu'à une facette spécifique de la technologie XML. Les règles et principes généraux énumérés ci-dessous fournissent le socle commun de l'élaboration de données et de documents aux fins de l'information sur la propriété intellectuelle.

- [GD-01] Tous les schémas XML DOIVENT reposer sur des spécifications techniques du Consortium W3C qui ont obtenu le statut de recommandation.
- [GD-02] Les schémas DOIVENT se conformer :  
à la première partie du schéma XML : Structures (<http://www.w3.org/TR/xmlschema-1/>);  
et  
à la deuxième partie du schéma XML : Types de données  
(<http://www.w3.org/TR/xmlschema-2/>).
- [GD-03] Les schémas DOIVENT utiliser la norme ISO/IEC 10646 – UCS – ensemble de caractères Unicode. La norme UTF-8 DOIT être utilisée pour coder les caractères Unicode.

### 2.2 Conventions de nommage XML

12. Ces conventions sont nécessaires pour assurer la cohérence, l'uniformité et l'exhaustivité du nommage et de la définition de toutes les ressources XML.

#### 2.2.1 Conventions de nommage de construction de schémas

13. Les conventions de nommage XML de la norme ST.96 de l'OMPI reposent sur les lignes directrices et principes décrits dans le document ISO 11179, Partie 5 – Principes de dénomination et d'identification. Le nom des types, éléments et attributs comprend les termes suivants :

- La classe objet désigne une activité ou un objet dans un contexte d'entreprise et représente le regroupement ou l'agrégation logique de données (dans un modèle logique de données) auquel une propriété appartient. La classe objet est désignée par le terme de classe objet.
- Le terme de propriété détermine les caractéristiques de la classe objet.
- Le terme de représentation classe le format de l'élément de donnée selon de grandes catégories. Les termes de représentation énumérés à l'appendice B du présent document devraient être utilisés pour la norme ST.96 de l'OMPI.
- Le terme de qualification est constitué d'un ou plusieurs mots qui aident à définir un élément de donnée et à le différencier d'autres éléments de donnée connexes et qui peuvent être associés à un terme de classe objet ou à un terme de propriété si cela s'avère nécessaire pour rendre un nom unique.

- [GD-04] Les noms de type, d'élément et d'attribut DOIVENT être composés de mots anglais, écrits selon l'orthographe anglaise figurant dans le dictionnaire Oxford English Dictionary. Les seules exceptions autorisées sont les sigles, les abréviations et d'autres mots tronqués qui sont énumérés à l'appendice C.
- [GD-05] Les noms de type, d'élément et d'attribut DEVRAIENT comprendre uniquement des noms, des adjectifs et des verbes au présent, à l'exception des sigles, abréviations et autres mots tronqués qui sont énumérés à l'appendice C.
- [GD-06] Les caractères utilisés dans des noms de types d'élément et d'attribut DOIVENT être limités à la série suivante : 'a-z, A-Z et 0-9'.
- [GD-07] La longueur maximale d'un nom d'élément NE DEVRAIT PAS être supérieure à 35 caractères. L'utilisation d'abréviations ou d'acronymes est encouragée pour réduire la longueur d'un nom d'élément. Toute nouvelle abréviation ou tout nouvel acronyme DOIT être ajouté(e) à l'appendice C.
- [GD-08] Les noms de type, d'élément et d'attribut DEVRAIENT être concis et s'expliquer d'eux-mêmes.

- [GD-09] Les noms d'élément DOIVENT utiliser les caractères majuscules de type "camel" (UCC). Par exemple, `CountryCode`.
- [GD-10] Les noms de type DOIVENT utiliser la forme UCC et avoir un type suffixe. Par exemple, `ApplicantType`.
- [GD-11] Les noms d'attribut DOIVENT utiliser les caractères minuscules de type "camel" (LCC). Par exemple, `currencyCode="EUR"`.
- [GD-12] Les sigles et les abréviations énumérés à l'appendice C DOIVENT toujours être utilisés en lieu et place du nom développé complet.
- [GD-13] Les sigles et abréviations au début d'une déclaration d'attribut DOIVENT tous figurer en minuscules. Par exemple, `biodepositDescriptionText`. Tous les autres sigles et abréviations utilisés dans une déclaration d'attribut DOIVENT figurer en majuscules comme indiqué dans l'appendice C.
- [GD-14] Les sigles et abréviations DOIVENT figurer comme indiqué dans l'appendice C pour les noms de type et d'élément.
- [GD-15] Les noms de type complexe DEVRAIENT inclure un terme de classe objet significatif.
- [GD-16] Lorsque nous avons besoin de deux termes de classe objet dans un nom de composante donnée compte tenu de la nature de l'entité opérationnelle, un type complexe d'association PEUT être utilisé. Les noms d'un type complexe d'association DEVRAIENT utiliser une structure de terme de classe objet du type complexe associant, un terme de propriété qui décrit la nature de l'association, et la classe objet du type complexe associé. Les termes de qualification peuvent précéder la classe objet et le terme de propriété. Par exemple, dans `ApplicantResidenceAddress`, `Applicant` est la classe objet de type complexe associant, `Residence` est la propriété et `Address` est la classe objet de type complexe associé.
- [GD-17] Les noms d'éléments basiques (ceux qui sont fondés sur un type simple ou complexe qui n'accepte pas les enfants) DEVRAIENT comporter un terme de classe objet, un terme de propriété et, le cas échéant, un terme de représentation. Les termes de qualification peuvent précéder le terme de classe objet et le terme de propriété.
- [GD-18] Un terme de classe objet DOIT toujours avoir la même signification sémantique dans un espace de nommage et PEUT comprendre plus d'un mot. Par exemple, `ContactInformation`.
- [GD-19] Le terme de propriété dans un nom DOIT être unique dans le contexte d'une classe objet mais PEUT être réutilisé dans différentes classes objets.
- [GD-20] Un terme de qualification PEUT être attaché à un terme de classe objet ou à un terme de propriété si cela s'avère nécessaire pour rendre un nom unique.
- [GD-21] Lorsqu'un nom comprend un terme de classe objet, un terme de propriété et un terme de représentation, le terme de classe objet DOIT précéder le terme de propriété et le terme de propriété DOIT précéder le terme de représentation. Un terme de qualification DEVRAIT précéder le terme associé de classe objet ou le terme de propriété.
- [GD-22] Si le terme de propriété finit avec le même mot que le terme de représentation (ou un mot équivalent), le terme de représentation DOIT être supprimé.
- [GD-23] Lorsqu'un terme de représentation est requis, les termes de représentation de l'appendice B DOIVENT être utilisés pour les termes de représentation dans les noms de composantes basiques.
- [GD-24] Dans un espace de nommage, tous les noms de type, d'élément et d'attribut DOIVENT être uniques.
- [GD-25] Le ou les mots dans un nom DEVRAIENT être utilisés au singulier sauf si le concept est lui-même au pluriel. Par exemple, `TotalMarkSeries`.
- [GD-26] Le nom d'un élément ou d'un type qui comprend une collection de composantes contextuellement reliées DEVRAIT avoir le suffixe "Bag". Par exemple, `EmailAddressBag` représente une collection d'éléments `EmailAddress`.

- [GD-27] Des mots qui relient comme "and", "of" et "the" NE DEVRAIENT PAS être utilisés dans les noms de type, d'élément et d'attribut à moins qu'ils ne fassent partie de la terminologie commerciale.
- [GD-28] Les noms de type, d'élément et d'attribut NE DOIVENT PAS être traduits, modifiés ou remplacés dans un but quel qu'il soit.
- [GD-29] Les noms de type et d'élément NE DOIVENT PAS faire référence à des numéros d'article et de règle. Par exemple, PCTRule702C pour le PCT.

### 2.2.2 Conventions de nommage des fichiers d'un schéma

14. Les noms de fichier d'un schéma et les noms d'un schéma sont souvent associés. Les noms de fichier d'un schéma reposent sur les noms de schéma correspondants. C'est ainsi, par exemple, que le nom de fichier PostalAddressType.xsd vient du nom de schéma PostalAddressType. Par conséquent, les conventions de nommage des fichiers de schéma sont liées aux règles des conventions de nommage XML dans le présent document.

15. Un fichier de schéma PEUT contenir les informations de la version. Un schéma qui en est au stade de projet peut être révisé. Les projets de schéma doivent être indiqués comme tels dans le nom de fichier du schéma, au moyen de la lettre "D" et du numéro de la révision.

- [GD-30] Les caractères utilisés dans les noms de fichier de schéma DOIVENT appartenir à la série suivante : 'a-z, A-Z, 0-9, caractère de soulignement "\_", et point ".".
- [GD-31] Un nom de fichier de schéma DOIT comporter deux parties obligatoires avec un délimiteur et les informations facultatives de la version avec deux délimiteurs, c'est-à-dire :  
<nom de la composante>{"\_" "V"<numéro de la version majeure>"\_"<numéro de la version mineure>"}."<extension du fichier>. Par exemple, EmailAddressType.xsd, languageCode.xsd, ApplicationBody\_V1\_0.xsd.
- [GD-32] Un nom de fichier du projet de schéma DOIT comporter quatre parties obligatoires avec deux délimiteurs et les informations facultatives de la version avec deux délimiteurs supplémentaires, c'est-à-dire : < nom de la composante>{"\_" "V"<numéro de la version majeure>"\_"<numéro de la version mineure>"}\_"D"<numéro de la révision>"}."<extension du fichier>, par exemple, Contact\_D3.xsd, TrademarkApplication\_V1\_1\_D1. Si un projet de schéma est fondé sur un schéma existant et que son nom de fichier inclut les informations de la version, les numéros des versions majeure et mineure dans le nom de fichier du projet de schéma DEVRAIENT être ceux qui sont indiqués dans le fichier de schéma sur lequel est fondé le projet de schéma. Si le projet de schéma est nouveau, le numéro de la version majeure dans le nom de fichier du projet de schéma DEVRAIT être celui qui est indiqué dans l'espace de nommage correspondant et un numéro de version mineure dans le fichier du projet de schéma DEVRAIT être zéro "0".

### *2.3 Stratégie de modularité*

16. La norme ST.96 de l'OMPI repose largement sur la modularité dans la conception du schéma. La division de la réalisation physique des schémas en plusieurs modules fournit un mécanisme au moyen duquel les composantes pouvant être réutilisées peuvent être importées à la place des schémas complets. C'est pourquoi cette norme recommande d'éviter la définition de tous les éléments et composantes logiques sous la forme d'un seul schéma XML monolithique, qui réduit la possibilité de partager et de réutiliser des éléments individuels ou des composantes logiques définis comme étant un groupe dans un schéma.

#### 2.3.1 Modules d'un schéma

17. Les composantes définies dans la norme ST.96 de l'OMPI sont classées en composantes communes, composantes brevets, composantes marques ou composantes dessins et modèles industriels. Les composantes communes DEVRAIENT être indépendantes du contexte (ou opérationnellement indépendantes) ou partagées par au moins deux types de titre de propriété intellectuelle. Exceptionnellement, compte tenu de l'évolution de la norme ST.96 dans l'avenir, certaines composantes telles que l'élément "P" pourraient être définies comme des composantes communes.

18. Il y a trois différents types de composante, à savoir de base, agrégées et document; les composantes document, constituées de composantes document de base, agrégées ou autres, sont définies pour répondre à un besoin opérationnel. Les composantes de base et agrégées sont définies dans le domaine commun. Les composantes de base, agrégées et document sont définies dans les domaines des brevets, des marques et des dessins et modèles.

Niveau	Description
Composante de base	Une composante de base s'entend d'un type de donnée W3C incorporé, d'un type simple ou d'un type complexe avec la définition <code>xsd:simpleContent</code> . Elle est représentée par un élément ou un attribut. Par exemple, <code>FirstName (string)</code> , <code>PhoneNumberCategory (PhoneNumberCategoryType)</code> .
Composante agrégée	Une composante agrégée est une collection de composantes de base connexes et/ou d'autres composantes agrégées qui, ensemble, ont une signification commerciale distincte indépendante d'un contexte commercial spécifique. Elle est représentée par un élément ou un type. Par exemple, <code>Name (NameType)</code> , <code>Contact (ContactType)</code> .
Composante document	Une composante document est un groupe de composantes comprenant des composantes de base, agrégées et/ou d'autres composantes document, qui a trait à un aspect spécifique d'une opération de propriété intellectuelle telle que le dépôt, l'examen, la publication, la délivrance ou l'échange de données. Par exemple, la composante <code>ApplicationBody</code> , fondée sur <code>ApplicationBodyType</code> , représente une description de brevet et peut être utilisée pour le dépôt et la publication d'une description de brevet.

### 2.3.2 Renvoi à des schémas externes

19. Chaque module de schéma a trait à un espace de nommage distinct. Dans la norme ST.96, chaque module de schéma est défini dans l'un des quatre espaces de nommage qui sont réservés aux modules du schéma de la composante commune, de la composante brevets, de la composante marques et de la composante dessins et modèles industriels respectivement.

20. Deux structures de schéma W3C, à savoir `include` et `import`, sont utilisées pour les renvois à des schémas externes. La structure `include` doit être utilisée lorsque les schémas incluant et inclus ont le même espace de nommage cible. Le terme schéma incluant s'entend du schéma qui inclut un schéma externe tandis que le terme schéma inclus s'entend du schéma externe. La structure `import` doit être utilisée lorsque les schémas incluant et inclus ont des espaces de nommage cibles différents. Cette technique peut être utile si une organisation ne doit pas limiter l'utilisation de structures dans des schémas uniquement à celles qui appartiennent à un espace de nommage particulier.

- [SD-01] Les modules du schéma de la composante brevets, de la composante marques, de la composante dessins et modèles industriels, de la composante indications géographiques et de la composante droit d'auteur DOIVENT utiliser la structure `xsd:import` pour renvoyer aux modules du schéma de la composante commune.
- [SD-02] Les modules du schéma de la composante commune NE DOIVENT PAS renvoyer aux modules du schéma de la composante brevets, de la composante marques, de la composante dessins et modèles industriels, de la composante indications géographiques ou de la composante droit d'auteur.
- [SD-03] Un module du schéma de la composante brevets, de la composante marques, de la composante dessins et modèles industriels, de la composante indications géographiques ou de la composante droit d'auteur PEUT renvoyer à des modules du schéma de la composante commune, à des schémas du module de la même composante et à des schémas approuvés de normes industrielles, mais NE DEVRAIT PAS renvoyer à des modules du schéma d'autres composantes, à une exception près : les schémas de la composante indications géographiques PEUVENT renvoyer à des schémas de la composante marques et inversement, car les secteurs d'activité en cause se recoupent partiellement.

## 2.4 Réutilisabilité

21. Le découplage et la cohésion sont d'importants principes de conception des schémas XML. Le découplage vise à minimiser les dépendances entre les éléments, aussi bien dans l'instance de document que dans le schéma. La cohésion vise à regrouper les morceaux connexes de données. Quelques types de styles de conception ont fait leur apparition et prennent en considération le découplage et la cohésion dans les schémas XML.

22. Les styles de conception répondent également à la question de savoir comment varie la granularité des composantes (éléments et types) à réutiliser. Le choix d'un style approprié est une étape essentielle de la conception de schémas. C'est pourquoi il faut le choisir avant de concevoir des schémas.

23. Les styles de conception les plus courants sont Russian Doll, Salami Slice, Venetian Blind et Garden of Eden. Les styles varient selon le nombre de leurs composantes globales (éléments ou types). Pour comprendre les styles de conception, il est nécessaire de faire la différence entre une composante globale et une composante locale. Une composante globale est un enfant immédiat de la structure du schéma dans le fichier de définition d'un schéma XML. Une composante locale n'est pas un enfant immédiat de la structure du schéma dans le fichier de définition d'un schéma XML. Les composantes globales sont associées à l'espace de nommage cible et peuvent être réutilisées dans d'autres schémas. Il est également important de comprendre qu'un élément défini dans l'espace de nommage global peut être la base d'une instance XML valide correspondant au schéma défini pour cet espace de nommage.

24. La plupart des schémas du monde réel font de Venetian Blind ou de Garden of Eden leur style de choix car ils sont les plus réutilisables. Aux fins du découplage et de la cohésion, Venetian Blind peut être meilleur que Garden of Eden mais, lorsqu'il s'agit de la réutilisabilité, Garden of Eden peut être meilleur que Venetian Blind en utilisant à la fois les types et les éléments globaux. La question la plus importante qui se pose pour une approche fondée sur un type (Venetian Blind) est le risque d'élaborer un vocabulaire d'élément incohérent où des éléments sont déclarés localement et peuvent être réutilisés sans tenir compte de la clarté sémantique et de la cohérence des types. C'est pourquoi la norme ST.96 de l'OMPI recommande d'adopter Garden of Eden comme style de conception.

- [SD-04] Les schémas existants DOIVENT être utilisés, le cas échéant, avant d'en créer de nouveaux.
- [SD-05] Les schémas DEVRAIENT utiliser dans toute la mesure du possible des éléments et types définis dans les schémas existants.
- [SD-06] Tous les types, éléments et attributs DOIVENT être déclarés globalement.
- [SD-07] Les schémas NE DOIVENT PAS utiliser la structure `xsd:redefine`.

## 2.5 Espaces de nommage

25. Les espaces de nommage sont utilisés pour identifier de manière unique les éléments et attributs ayant le même nom lorsqu'ils sont combinés dans un seul document. Ils associent les structures de schéma à un espace conceptuel qui définit un vocabulaire de balisage. Une instance XML peut contenir des noms de plus d'un vocabulaire XML. Si un espace de nommage est donné à chaque vocabulaire, il est possible de résoudre le problème d'ambiguïté entre des éléments ou attributs portant le même nom. Les espaces de nommage doivent être uniques et rémanents. Dans la norme ST.96 de l'OMPI, il est recommandé d'utiliser une configuration à espaces de nommage multiples.

- [SD-08] Les schémas DOIVENT utiliser des espaces de nommage.
- [SD-09] Les déclarations publiées des espaces de nommage NE DEVRAIENT PAS être modifiées.

### 2.5.1 Déclaration et qualification des espaces de nommage

- [SD-10] Les schémas DOIVENT déclarer l'espace de nommage du Consortium W3C.
- [SD-11] Les schémas DOIVENT utiliser les qualifications des espaces de nommage pour toutes les structures de schéma du Consortium W3C.
- [SD-12] Les schémas DOIVENT utiliser l'espace de nommage formaté en URI.
- [SD-13] Tous les schémas DOIVENT avoir des attributs `elementFormDefault` et `attributeFormDefault` avec la valeur "qualified" dans l'élément de base `xsd:schema`.

26. Pour des raisons d'efficacité, les offices de propriété intellectuelle peuvent décider de réduire ou même d'éliminer les espaces de nommage dans les systèmes de traitement XML. Néanmoins, concernant l'échange, les règles ci-dessus doivent être observées.

### 2.5.2 Espaces de nommage dans le schéma XML

27. La norme ST.96 définit quatre espaces de nommage : commun, de brevets, de marques, de dessins et modèles. Ces espaces de nommages sont symboliques et constants dans le temps.

28. L'exemple suivant présente un espace de nommage bien formé :

```
<?xml version="1.0" encoding="UTF-8"?>
<xsd:schema xmlns:com="http://www.wipo.int/standards/XMLSchema/ST96/Common"
xmlns:xsd="http://www.w3.org/2001/XMLSchema">
.....
</xsd:schema>
```

[SD-14] Les noms des espaces de nommage DOIVENT avoir la structure suivante :  
<http://www.wipo.int/standards/XMLSchema/ST96/<category>>; où  
<category> est un symbole identifiant le domaine du schéma utilisé : Common, Patent, Trademark, Design, Geographical Indication or Copyright.

[SD-15] Les schémas DEVRAIENT utiliser "xsd" comme un préfixe d'espace de nommage pour toutes les structures de schéma du Consortium W3C, "com" pour tous les schémas de la composante commune, "pat" pour tous les schémas de la composante brevets, "tmk" pour tous les schémas de la composante marques, "dgn" pour tous les schémas de la composante dessins et modèles industriels, "gin" pour les schémas de la composante indications géographiques, "cpy" pour les schémas de la composante droit d'auteur, "tbl" pour les schémas de la composante OASIS Table et "mathml" pour les schémas de la composante MathML.

### 2.5.3 Espaces de nommage cibles

29. La déclaration d'un espace de nommage cible dans un schéma permet de faire en sorte que toutes les structures à l'intérieur du schéma soient associées à un espace de nommage. Inversement, sans un espace de nommage cible, les structures déclarées dans le schéma n'appartiendraient à aucun espace de nommage. Un schéma peut certes avoir plus d'un espace de nommage déclaré mais un espace de nommage seulement peut être désigné comme l'espace de nommage cible. Dans les recommandations du Consortium W3C, il n'est pas requis qu'un espace de nommage cible soit déclaré dans un schéma. Il est cependant recommandé d'utiliser un espace de nommage cible dans la norme ST.96 de l'OMPI.

30. Un schéma déclare un espace de nommage cible qui correspond à l'un des espaces de nommage déclarés. Conformément à l'espace de nommage associé, les schémas doivent déclarer un des quatre espaces de nommage cibles suivants qui sont définis dans la norme ST.96 de l'OMPI :

- <http://www.wipo.int/standards/XMLSchema/ST96/Common>, par exemple,

```
<?xml version="1.0" UTF-8?>
<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:com="http://www.wipo.int/standards/XMLSchema/Common"
targetNamespace="http://www.wipo.int/standards/XMLSchema/Common"
elementFormDefault="qualified" attributeFormDefault="qualified"
version="V4_0">
...
</xsd:schema>
```

- <http://www.wipo.int/standards/XMLSchema/ST96/Patent>, par exemple,

```
<?xml version="1.0" UTF-8?>
<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:pat="http://www.wipo.int/standards/XMLSchema/ST96/Patent"
targetNamespace="http://www.wipo.int/standards/XMLSchema/ST96/Patent"
elementFormDefault="qualified" attributeFormDefault="qualified"
version="V4_0">
<xsd:include schemaLocation="xxx.xsd"/>
</xsd:schema>
```

- <http://www.wipo.int/standards/XMLSchema/ST96/Trademark>, par exemple,

```
<?xml version="1.0" UTF-8?>
<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:tmk="http://www.wipo.int/standards/XMLSchema/ST96/Trademark"
targetNamespace="http://www.wipo.int/standards/XMLSchema/ST96/Trademark"
elementFormDefault="qualified" attributeFormDefault="qualified"
version="V4_0">
...
</xsd:schema>
```

- <http://www.wipo.int/standards/XMLSchema/ST96/Design>, par exemple,

```
<?xml version="1.0" UTF-8?>
<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:dgn="http://www.wipo.int/standards/XMLSchema/ST96/Design"
targetNamespace="http://www.wipo.int/standards/XMLSchema/ST96/Design"
elementFormDefault="qualified" attributeFormDefault="qualified"
version="V4_0">
...
</xsd:schema>
```

- [SD-16] Chaque schéma DOIT déclarer l'espace de nommage cible à l'aide de l'attribut `xsd:targetNamespace`.
- [SD-17] Le `targetNamespace` du schéma DOIT correspondre au nom de l'espace de nommage d'un des espaces de nommage déclarés mais pas à l'espace de nommage du Consortium W3C.
- [SD-18] Les schémas de la composante commune DOIVENT utiliser <http://www.wipo.int/standards/XMLSchema/ST96/Common> comme espace de nommage cible.
- [SD-19] Les schémas de la composante brevets DOIVENT utiliser <http://www.wipo.int/standards/XMLSchema/ST96/Patent> comme espace de nommage cible.
- [SD-20] Les schémas de la composante marques DOIVENT utiliser <http://www.wipo.int/standards/XMLSchema/ST96/Trademark> comme espace de nommage cible.
- [SD-21] Les schémas de la composante dessins et modèles industriels DOIVENT utiliser <http://www.wipo.int/standards/XMLSchema/ST96/Design> comme espace de nommage cible.
- [SD-65] Les schémas de la composante indications géographiques DOIVENT utiliser <http://www.wipo.int/standards/XMLSchema/ST96/GeographicalIndication> comme espace de nommage cible.
- [SD-66] Les schémas de la composante droit d'auteur DOIVENT utiliser <http://www.wipo.int/standards/XMLSchema/ST96/Copyright> comme espace de nommage cible.

#### 2.5.4 Espaces de nommage implicites

31. Un espace de nommage implicite réduit la verbosité dans un schéma. Toutefois, la déclaration d'un espace de nommage implicite dans un schéma accroît l'ambiguïté car l'omission de préfixes rend plus difficile l'identification de l'espace de nommage auquel une structure appartient. L'utilisation d'espaces de nommage implicites peut causer la coercition de ces espaces, ne permettant pas de discerner l'origine des structures du schéma en examinant le schéma incluant. Par conséquent, l'utilisation d'un espace de nommage implicite n'est pas recommandée dans la norme ST.96 de l'OMPI.

[SD-22] Les schémas NE DEVRAIENT PAS utiliser un espace de nommage implicite.

#### *2.6 Versions de schéma*

32. La rétrocompatibilité est une question qui se pose lorsque des schémas existants doivent être modifiés. En général, il y a rétrocompatibilité si une instance XML qui valide une version antérieure continue de valider la nouvelle version du schéma.

##### 2.6.1 Modifications majeures et mineures

33. Les modifications apportées aux schémas sont appelées modifications majeures et modifications mineures. Les modifications majeures sont celles qui ne sont pas rétrocompatibles avec des versions de schémas antérieures alors que les modifications mineures le sont. Les nouvelles versions mineures d'un schéma DOIVENT pouvoir valider les instances de document créées avec des versions mineures antérieures de ce schéma, avec la même version majeure. Toutefois, il ne faut pas s'attendre à ce que les instances de document valident des versions d'un schéma précédant celle avec laquelle ils ont été créés. En outre, le schéma peut faire l'objet de révisions s'il en est au stade de projet.

###### 2.6.1.1 Versions majeures

34. La version majeure d'un schéma constitue une version ayant fait l'objet de modifications significatives ou non rétrocompatibles. Si une instance XML fondée sur une version majeure plus ancienne d'un schéma cherche une validation en fonction de la nouvelle version, il risque de connaître des erreurs de validation. Une nouvelle version majeure sera établie lorsque des modifications significatives ou non rétrocompatibles ont lieu, comme par exemple :

- élimination ou modification des valeurs dans les énumérations;
- modification des noms d'élément, de type et d'attribut;
- suppression ou ajout d'éléments ou d'attributs obligatoires; et
- modification de la cardinalité qui, de facultative, devient obligatoire.

###### 2.6.1.2 Versions mineures

35. Dans la version majeure d'un schéma de la norme ST.96 de l'OMPI, il peut y avoir une série de modifications mineures ou rétrocompatibles. La version mineure d'un schéma ST.96 détermine sa compatibilité avec des versions mineures antérieures et ultérieures de la même version majeure. La version mineure aide donc à établir la rétrocompatibilité et la compatibilité ultérieure. Les numéros des versions mineures ne seront augmentés que lorsque des modifications compatibles ont lieu, par exemple :

- ajout de valeurs aux énumérations,
- extensions facultatives, et
- ajout d'éléments et/ou d'attributs facultatifs

[SD-23] Les nouvelles versions mineures d'un schéma DOIVENT pouvoir valider toutes les instances de document créées avec des versions mineures antérieures de ce schéma avec la même version majeure.

[SD-24] La version majeure d'un schéma DOIT être renforcée si le nouveau schéma n'est pas en mesure de valider les instances de document XML existantes avec la version majeure utilisée.

[SD-25] *Supprimé*

- [SD-26] *Supprimé*
- [SD-27] Le numéro de la version majeure ou mineure d'un schéma DEVRAIT être modifié lorsqu'un schéma inclus ou importé est actualisé.
- [SD-28] Lorsque la structure d'un schéma est altérée, tous les schémas dans cette version DOIVENT faire l'objet de la même augmentation de numéro de la version.
- [SD-29] Dans la création d'un nouveau schéma, les versions les plus récentes de tous les schémas inclus ou importés DEVRAIENT être utilisées.

### 2.6.2 Stratégie de contrôle de version du schéma

36. Les conventions ci-après définissent ensemble la stratégie de contrôle des schémas :

- a) structure de fichier de publication des schémas et conventions de nommage des fichiers des schémas;
- b) utilisation de l'attribut `version` du schéma XML incorporé pour toutes les composantes; et
- c) utilisation d'un attribut `schemaVersion` défini par l'utilisateur au niveau d'une instance de document; et
- d) utilisation d'un attribut `ipoVersion` défini par l'utilisateur au niveau d'une instance de document.

#### 2.6.2.1 Dossier et conventions de nommage des fichiers dans le contrôle de version du schéma

37. Les informations sur la version DEVRAIENT être incorporées dans le dossier, le nom de fichier du schéma au niveau du document et le nom de fichier du schéma mis à plat. D'après les règles de nommage des fichiers définies dans le présent document, le dossier de diffusion du schéma, le nom de fichier du schéma au niveau du document et le nom de fichier mis à plat contiennent un numéro de version majeure et un numéro de version mineure. Dans la structure du dossier de diffusion du schéma de la norme ST.96, le numéro de la version de diffusion est suivi des types de composante, à savoir communes, brevets, marques, dessins et modèles et normes externes. Par exemple, la version 3.0 de la composante `Contact` figure dans le dossier `ST96/V3_0/Common`.

- [SD-64] Le dossier de diffusion du schéma, un nom de fichier de schéma au niveau d'un document et un nom de fichier de schéma mis à plat DOIVENT inclure les informations de la version correspondantes, y compris le numéro de la version majeure et celui de la version mineure.

#### 2.6.2.2 Attribut `version` du schéma XML incorporé dans le contrôle de version du schéma

38. La spécification d'un schéma W3C permet la définition d'un attribut `version` dans l'élément de base d'un schéma. Dans la norme ST.96 de l'OMPI, les schémas doivent inclure aussi bien le numéro de la version majeure que le numéro de la version mineure pour chaque fichier de schéma utilisant l'attribut de la `version` du schéma W3C.

Par exemple,

```
<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:pat="http://www.wipo.int/standards/XMLSchema/ST96/Patent"
targetNamespace="http://www.wipo.int/standards/XMLSchema/ST96/Patent"
elementFormDefault="qualified" attributeFormDefault="unqualified" version="
V4_0">
...
</xsd:schema>
```

- [SD-30] L'attribut `version` du schéma W3C DOIT comprendre à la fois le numéro de la version majeure et le numéro de la version mineure pour chaque fichier de schéma dans le format suivant : "V"<major version number>"\_ "<minor version number>.

2.6.2.3 Attributs `st96Version` et `ipoVersion` définis par l'utilisateur dans le contrôle de version du schéma pour les instances XML

39. L'attribut `st96Version` défini par l'utilisateur est nécessaire pour s'assurer qu'un document donné se réfère directement à une version spécifique du schéma XML de la norme ST.96. Cet attribut définira clairement la version du schéma de la norme ST.96 que cible le document ou un schéma XML appliqué auquel il renvoie. Cet attribut devrait être déclaré nécessaire avec une valeur fixe.

40. L'attribut `ipoVersion` défini par l'utilisateur est nécessaire pour s'assurer qu'un document donné se réfère directement à une version spécifique d'un schéma appliqué par un office de propriété intellectuelle. Cet attribut définira clairement la version du schéma de l'office de propriété intellectuelle que cible le document. Cet attribut devrait être déclaré facultatif et ne devrait pas être renseigné lorsque les composantes du schéma de la norme ST.96 sont utilisées sans modification.

41. Compte tenu de la rétrocompatibilité, de la compatibilité ultérieure et des échanges de données entre les offices de propriété intellectuelle pour lesquels des versions mineures différentes sont utilisées, le numéro de la version majeure et celui de la version mineure doivent être renseignés. Les exemples ci-après indiquent comment les attributs peuvent être définis dans le schéma XML de la norme ST.96, le schéma d'application de l'office et l'instance XML.

42. L'exemple ci-après montre comment les attributs de `st96Version` et de `ipoVersion` devraient être définis dans le schéma de la norme ST.96 et dans le schéma d'application de l'office de propriété intellectuelle; et utilisés dans une instance XML :

`st96Version` de la version 4.0 de la norme ST.96 :

```
<?xml version="1.0" encoding="UTF-8"?>
<xsd:schema xmlns:com="http://www.wipo.int/standards/XMLSchema/ST96/Common"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
targetNamespace="http://www.wipo.int/standards/XMLSchema/ST96/Common"
elementFormDefault="qualified" attributeFormDefault="qualified"
version="V4_0">
  <xsd:attribute name="st96Version" type="xsd:token" fixed="V4_0">
    ..
  </xsd:attribute>
</xsd:schema>
```

ipoVersion de la version 4.0 de la norme ST.96 :

```
<?xml version="1.0" encoding="UTF-8"?>
<xsd:schema xmlns:com="http://www.wipo.int/standards/XMLSchema/ST96/Common"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
targetNamespace="http://www.wipo.int/standards/XMLSchema/ST96/Common"
elementFormDefault="qualified" attributeFormDefault="qualified"
version="V4_0">
  <xsd:attribute name="ipoVersion" type="xsd:token">
    ...
  </xsd:attribute>
</xsd:schema>
```

ApplicationBodyType de la version 4.0 de la norme ST.96 :

```
<?xml version="1.0"?>
<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema"
targetNamespace="http://www.wipo.int/standards/XMLSchema/ST96/Patent"
xmlns:com="http://www.wipo.int/standards/XMLSchema/ST96/Common"
xmlns:pat="http://www.wipo.int/standards/XMLSchema/ST96/Patent"
elementFormDefault="qualified" attributeFormDefault="qualified"
version="V4_0">
<xsd:import namespace="http://www.wipo.int/standards/XMLSchema/ST96/Common"
schemaLocation="../../Common/st96Version.xsd"/>
<xsd:import namespace="http://www.wipo.int/standards/XMLSchema/ST96/Common"
schemaLocation="../../Common/ipoVersion.xsd"/>
.....
<xsd:complexType name="ApplicationBodyType">
  .....
  <xsd:attribute ref="com:st96Version" use="required" />
<xsd:attribute ref="com:ipoVersion"/>
  .....
</xsd:complexType>
</xsd:schema>
```

ipoVersion et ApplicationBodyType de la version 1.0 du schéma d'application de l'USPTO, fondé sur l'élément ApplicationBodyType de la version 3.0 de la norme ST.96 ci-dessus :

```
<?xml version="1.0" encoding="UTF-8"?>
...
  <xsd:attribute name="ipoVersion" type="xsd:token" fixed="US_V1_0">
    ...
  </xsd:attribute>
...
```

```
<?xml version="1.0"?>
<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:uspat="urn:us:gov:doc:uspto:patent"
targetNamespace="urn:us:gov:doc:uspto:patent"
xmlns:com="http://www.wipo.int/standards/XMLSchema/ST96/Common"
xmlns:pat="http://www.wipo.int/standards/XMLSchema/ST96/Patent"
elementFormDefault="qualified" attributeFormDefault="qualified"
version="V4_0">
<xsd:import namespace="http://www.wipo.int/standards/XMLSchema/ST96/Common"
schemaLocation="../../Common/st96Version.xsd"/>
<xsd:import namespace="http://www.wipo.int/standards/XMLSchema/ST96/Common"
schemaLocation="../../Common/ipoVersion.xsd"/>
.....
<xsd:complexType name="ApplicationBodyType">
.....
  <xsd:attribute ref="com:st96Version" use="required" />
<xsd:attribute ref="com:ipoVersion" />
.....
</xsd:complexType>
</xsd:schema>
```

Une instance XML fondée sur l'élément `ApplicationBody` de la version 1.0 du schéma d'application de l'USPTO :

```
<?xml version="1.0" encoding="UTF-8"?>
<uspat: ApplicationBody xmlns:xsi="http://www.w3.org/2001/XMLSchema-
instance"
xmlns:com="http://www.wipo.int/standards/XMLSchema/ST96/Common"
xmlns:pat="http://www.wipo.int/standards/XMLSchema/ST96/Patent"
xmlns:uspat="urn:us:gov:doc:uspto:patent"
xmlns:uscom="urn:us:gov:doc:uspto:common" "
xsi:schemaLocation="urn:us:gov:doc:uspto:patent
PE2E/V1_0/USPatent/Document/ApplicationBody_V1_0.xsd" com:st96Version="V4_0"
com:ipoVersion="US_V1_0">
.....
</uspat:ApplicationBody>
```

[SD-31] Le schéma pour une composante document DOIT déclarer un attribut nécessaire appelé `st96Version` dans l'élément de base avec une valeur fixe correspondant à la version de diffusion du schéma XML de la norme ST.96, dans le format suivant : "V"<major version number>"\_ "<minor version number>.

[SD-63] Le schéma pour une composante document DOIT déclarer un attribut facultatif appelé `ipoVersion` dans l'élément de base avec une valeur fixe correspondant à la version de diffusion du schéma d'application de l'office de propriété intellectuelle, dans le format suivant : <ST.3 Code>"\_ "V"<major version number>"\_ "<minor version number>, par exemple, "US\_V2\_0".

## 2.7 Possibilités de conversion avec d'autres normes XML de l'OMPI

43. Avant l'adoption de la norme ST.96 de l'OMPI, les normes ST.36, ST.66 et ST.86 avaient déjà été utilisées par les offices de propriété intellectuelle, c'est pourquoi il est essentiel que la norme ST.96 de l'OMPI maintienne les possibilités de conversion entre les normes ST.96 et ST.36, ST.66 ou ST.86. Pour veiller à ce que les données puissent être traitées de manière satisfaisante afin de répondre aux besoins commerciaux des offices de propriété intellectuelle et des fournisseurs d'information en matière de propriété intellectuelle, la norme ST.96 cherche à établir le degré nécessaire de possibilités de conversion avec les normes ST.36, ST.66 et ST.86 de l'OMPI.

44. Par possibilités de conversion, on entend que l'ensemble des unités d'information atomiques (niveau le plus bas des éléments de granularité et des branches terminales de l'attribut, généralement un élément sans aucun élément enfant) qui pourraient être incorporées dans une instance XML conforme à la norme ST.96 peut être converti en unités d'information atomiques pouvant être incorporées dans une instance XML conforme aux normes ST.36, ST.66 ou ST.86, et inversement.

45. Les listes d'énumération devraient tenir compte des valeurs prévues pour les unités d'information atomiques respectives figurant dans les normes ST.36, ST.66 ou ST.86 de l'OMPI. Elles devraient inclure les valeurs, si nécessaire, "Autre", ce qui signifie que la valeur fournie dans les données d'origine n'est pas présente dans la liste énumérée, et "inconnue", ce qui signifie que l'information était absente ou incorrecte dans les données d'origine (instances XML conformes aux normes ST.36, ST.66 et ST.86). Par habitude, la valeur "non définie" est utilisée à la place de "inconnue" dans certaines listes d'énumération.

[SD-32] Les schémas ST.96 DEVRAIENT être élaborés avec le même niveau de cardinalité et de granularité afin de faciliter les possibilités de conversion des instances ST.96 avec les instances ST.36, ST.66 ou ST.86.

[SD-33] Les énumérations contenues dans les schémas NE DEVRAIENT PAS inclure les valeurs "Autre" ou "Inconnue" à moins qu'elles ne soient nécessaires.

## 2.8 Schémas de normes industrielles

46. Lorsque cela s'applique au contenu d'un document, c'est-à-dire lorsque le contenu n'est pas spécifique au domaine de la propriété intellectuelle, des schémas de normes industrielles devraient être utilisés. Les schémas de normes industrielles recommandés dans le présent document sont le schéma de tableau XML OASIS et MathML.

47. Le premier, disponible à l'adresse suivante : <http://www.oasis-open.org/docbook/xmlschema/1.0b1/calstbl.xsd>, est un schéma type qui doit être précisé davantage pour répondre aux besoins commerciaux. OASISTable.xsd est défini dans le schéma XML de la norme ST.96 sur la base du schéma de tableau XML OASIS.

48. En ce qui concerne les formules mathématiques, il est recommandé d'utiliser MathML, version 3. Pour de plus amples informations sur MathML, version 3, voir à l'adresse suivante : <http://www.w3.org/TR/MathML3>.

49. En ce qui concerne les formules ou structures chimiques, il n'existait aucune recommandation du Consortium W3C ni aucune norme industrielle XML largement admise lors de l'élaboration du présent document. Il y a, dans la réalité, un grand nombre de normes de balisage XML pour différents types chimiques mais aucune ne peut être considérée comme étant universellement acceptée. Par conséquent, le présent document ne recommande aucune norme XML particulière pour les données chimiques.

[SD-34] MathML, version 3, DEVRAIT être utilisée pour les formules mathématiques.

[SD-35] OASISTable.xsd, qui repose sur le schéma de tableau d'échange XML OASIS, DEVRAIT être utilisé pour les tableaux.

[SD-36] Les schémas de normes industrielles DEVRAIENT être incorporés par renvoi uniquement et après l'approbation préalable du Comité des normes de l'OMPI.

## 3. CONVENTIONS DE CONSTRUCTION DE SCHÉMAS XML

### 3.1 Définitions des types

50. Les types représentent le genre d'information que peuvent détenir les éléments et les attributs comme, par exemple, les chaînes de caractères ou les dates. Les types et les éléments sont souvent associés.

#### 3.1.1 Types simples

51. L'utilisation de types simples améliore la qualité des données dans les applications XML car toutes les applications qui utilisent des types simples sont soumises aux mêmes validations par des processeurs XML. Lorsque le format lexical d'un type simple ne convient pas, les développeurs de schémas peuvent créer leurs propres types de données à l'aide de la syntaxe d'expression régulière du schéma W3C. Les types simples comprennent les types de données incorporés du Consortium W3C et les types de données définies par l'utilisateur.

##### 3.1.1.1 Types de données incorporées du W3C

52. Les types de données incorporées du W3C sont les types de données qui ont été définies par le Consortium W3C et incorporées dans la norme du schéma W3C comme, par exemple, `date`, `Boolean`, `string` et `token`.

### 3.1.1.2 Types de données définies par l'utilisateur

53. L'un des avantages des schémas XML est la capacité qu'ils offrent de créer des types de données définies par l'utilisateur. Ceux-ci reposent sur les types de données incorporées du W3C et peuvent également être tirés des types de données existants définis par l'utilisateur. Ils peuvent être tirés d'une des trois manières suivantes : restriction, list et union.

- [SD-37] Les schémas DEVRAIENT dans toute la mesure du possible utiliser des types simples.
- [SD-38] Des listes de codes DEVRAIENT être indiquées comme des listes d'énumération de type simple.
- [SD-39] Une liste de codes PEUT être indiquée comme une union de types simples.
- [SD-40] La norme ST.3 de l'OMPI (codes à deux lettres) DOIT être utilisée pour la représentation des offices de propriété intellectuelle ainsi que pour les pays/organisations désignés et prioritaires. Par exemple, `PriorityCountryCode="EP"`.
- [SD-41] La norme ISO 3166-1-Alpha-2 Éléments de code (codes de pays à deux lettres) DOIT être utilisée pour la représentation des noms de pays, des dépendances et autres zones d'intérêt géopolitique particulier, sur la base des listes de noms de pays provenant des Nations Unies.
- [SD-42] La norme ISO 639-1 (codes de langue à deux lettres) DOIT être utilisée pour les codes des langues.
- [SD-43] Les schémas DOIVENT indiquer les éléments et les attributs pour les valeurs de date et d'heure en utilisant les types de données relatives à la date et à l'heure du schéma XML du W3C.
- [SD-44] La norme ISO 4217-Alpha (codes pour la représentation des monnaies à trois lettres) DOIT être utilisée pour les codes des monnaies.
- [SD-45] Les caractères utilisés dans les énumérations DOIVENT se limiter aux caractères suivants : 'a-z, A-Z, 0-9, espace " " et caractère de soulignement "\_".
- [SD-46] Les valeurs d'énumération DEVRAIENT être suffisantes d'un point de vue sémantique et en anglais, et utiliser un nombre minimum de caractères. Les valeurs DEVRAIENT être tirées du langage utilisé couramment dans les activités relatives à la propriété intellectuelle.

### 3.1.2 Types complexes

54. Les types complexes sont des types définis par l'utilisateur qui contiennent des éléments ou attributs enfants.

- [SD-47] Des types complexes abstraits PEUVENT être utilisés.

## 3.2 Éléments et attributs

55. Les éléments sont le socle d'un document XML et ils sont représentés par des balises. Les attributs sont les structures du schéma W3C associées à des éléments qui fournissent des informations additionnelles sur les éléments.

### 3.2.1 Éléments ou attributs

56. L'une des principales décisions à prendre en matière de conception de schéma consiste à déterminer s'il faut représenter un élément de donnée sous la forme d'un élément ou d'un attribut XML. Les éléments peuvent être considérés comme contenant des données et les attributs comme contenant des métadonnées. Une fois qu'un élément de donnée a été converti en attribut, il n'est pas possible de l'étoffer davantage, c'est pourquoi les attributs DEVRAIENT être utilisés uniquement pour décrire des informations qui ne peuvent pas être étoffées ou sous-divisées ou qui ne le seront pas. Les schémas DEVRAIENT être conçus de telle sorte que les éléments soient les principaux détenteurs du contenu de l'information en matière de propriété intellectuelle dans les instances XML. Les attributs DEVRAIENT contenir des métadonnées accessoires, c'est-à-dire des données simples qui fournissent plus d'informations sur l'élément.

- [SD-48] Les schémas DEVRAIENT uniquement utiliser des attributs pour définir des données non commerciales. Par exemple, il est autorisé d'utiliser un attribut intitulé `sequenceNumber` pour les numéros d'ordre.

### 3.2.2 Éléments

#### 3.2.2.1 Cardinalité des éléments

57. Le terme cardinalité est défini comme le nombre d'éléments dans un ensemble.
58. La cardinalité est indiquée dans un schéma à l'aide des contraintes `minOccurs` et `maxOccurs` dans une déclaration d'éléments; ces contraintes sont également appelées indicateurs d'occurrence. Les indicateurs d'occurrence ne peuvent pas figurer dans les déclarations générales d'élément.

- [SD-49] Un indicateur d'occurrence NE DEVRAIT PAS être utilisé pour préciser une restriction qui est l'élément implicite dans un schéma. Par exemple, `minOccurs="1"` et `maxOccurs="1"` NE DEVRAIENT PAS être utilisés mais `minOccurs="2"` et `maxOccurs="3"` sont acceptables.

#### 3.2.2.2 Éléments vides

59. Un élément vide est un élément sans contenu texte, sans élément enfant et sans attribut. En général, l'absence d'un élément dans un schéma XML n'a aucune signification particulière; elle peut indiquer que l'information n'est pas connue ou qu'elle n'est pas applicable, ou que l'élément peut être absent pour une autre raison.
60. Bien que la norme ST.36 exploite les éléments vides comme étant un type d'indicateur `Boolean` qu'une condition était vraie ou fausse, en fonction de la présence ou de l'absence de l'élément, cette pratique est retirée dans la norme ST.96. Au contraire, il devrait être créé des éléments qui établissent explicitement la vraie condition ou la fausse, éliminant tous les doutes et l'ambiguïté pour les appareils de lecture ou les lecteurs humains.

- [SD-50] Les éléments vides NE DOIVENT PAS être définis dans les schémas, sauf pour le saut de ligne.

### 3.2.3 Attributs

61. Les attributs sont des structures de schéma W3C associées à des éléments qui fournissent des informations additionnelles sur les éléments. À la différence des éléments, les attributs ne peuvent pas être couplés en ce sens qu'il n'y a pas de sous-attributs : les attributs ne peuvent pas être étendus comme les éléments.
62. La cardinalité pour les attributs est différente de celle pour les éléments. L'indicateur `use` peut être déterminé pour un attribut avec l'une des valeurs suivantes : "required", "optional" ou "prohibited".

- [SD-51] L'attribut `use` dans une utilisation d'attribut DEVRAIT être omis lorsque l'attribut visé est facultatif car l'élément implicite est `use="optional"`.

### 3.2.4 Regroupement des éléments et attributs

63. Les compositeurs sont les structures de schéma W3C qui regroupent les déclarations d'éléments. Il y a trois catégories de compositeurs dans la norme du schéma W3C, à savoir `sequence`, `choice` et `all`.
64. Le compositeur `sequence` indique que les éléments qui y sont déclarés à l'intérieur doivent figurer dans une instance XML dans l'ordre déclaré. Il permet l'application de l'ordre des éléments.
65. Le compositeur `choice` indique qu'un seul des éléments qui y sont déclarés à l'intérieur peut figurer dans une instance XML.
66. Il arrive parfois que les indicateurs d'occurrence dans les compositeurs `sequence` et `choice` offrent la possibilité d'utiliser un compositeur `sequence/choice` comme, par exemple, un compositeur `sequence/choice` facultatif ou multiple, pour éviter la définition d'un niveau supplémentaire.

67. Le compositeur `all` indique que les éléments qui y sont déclarés à l'intérieur ne peuvent figurer qu'une seule fois dans une instance XML, dans n'importe quel ordre. Avec le compositeur `all`, aucun élément qui se trouve à l'intérieur ne peut figurer plus d'une fois et tous les éléments sont facultatifs si l'attribut `minOccurs` est fixé à 0. Dans le cas de l'échange de données, un élément au moins doit être obligatoire. L'utilisation du compositeur `all` n'est par conséquent pas recommandée pour les schémas.

[SD-52] Les schémas NE DEVRAIENT PAS utiliser le compositeur `all`.

[SD-53] Les schémas PEUVENT utiliser des indicateurs d'occurrence dans les compositeurs `xsd:sequence` et `xsd:choice`.

### 3.3 *Extension et restriction*

68. Certaines techniques autorisées dans les recommandations du Consortium W3C peuvent avoir une incidence sur l'harmonisation des données car elles confèrent de la souplesse à la structuration des données. Il s'agit des techniques suivantes : extension, restriction et groupes de substitution.

#### 3.3.1 Extension

69. L'extension de types complexes n'influe pas sur l'harmonisation des données alors que l'extension de types simples le fait car des composantes ajoutées ne seront pas échangées. Cependant, le fait de tirer des types simples en utilisant `union` ou `list` entrave l'interopérabilité car il permet de nouvelles valeurs qui ne sont pas traitées par le format d'échange de données.

[SD-54] Un schéma NE DOIT PAS contenir un type complexe avec `xsd:any` pour l'extensibilité.

#### 3.3.2 Restriction

70. Dans le contexte de l'harmonisation des données, la restriction de types complexes modifie la structure et peut donner lieu à des problèmes de compatibilité qui bloquent l'échange de données. Dans le cas de types simples, la restriction modifie les valeurs; aucun problème n'est attendu lorsque des données sont échangées.

#### 3.3.3 Groupes de substitution

71. Les groupes de substitution permettent à un élément global de remplacer un autre élément global dans une instance XML sans devoir apporter des modifications additionnelles au schéma. Ils n'encouragent toutefois pas l'harmonisation des noms d'élément. L'harmonisation est la clé de l'échange de données interopérables et l'utilisation des groupes de substitution est incompatible avec l'harmonisation.

[SD-55] Les schémas NE DOIVENT PAS utiliser les groupes de substitution.

### 3.4 *Contraintes en matière d'identité*

72. Comme dans tout système de stockage, un document XML doit offrir des moyens d'identifier des éléments d'information qu'il contient ou de s'y référer. Les deux caractéristiques qui permettent au XML de le faire avec un schéma XML du Consortium W3C sont `xsd:key/unique/keyref` et `xsd:ID/IDREF/IDREFS`.

73. Pour les renvois dans le document XML concerné, `xsd:ID/IDREF/IDREFS` PEUT coexister avec `xsd:key/unique/keyref` dans la même instance XML. La caractéristique `xsd:key/unique/keyref` DEVRAIT être utilisée pour toute composante dans un domaine où l'unicité doit demeurer complètement. Les caractéristiques `xsd:ID`, `xsd:IDREF` et `xsd:IDREFS` DEVRAIENT être utilisées dans les instances XML lorsque l'unicité et le renvoi à l'unicité sont requis pour l'ensemble du document. Les attributs `id`, `idref` et `idrefs` sont définis respectivement dans l'espace de nommage commun par `xsd:ID`, `xsd:IDREF` et `xsd:IDREFS`. L'attribut `idrefs` DEVRAIT être utilisé lorsque des renvois multiples sont attendus, par exemple dans les revendications.

74. Pour les renvois externes au document XML concerné, l'attribut `extRef` défini comme `xsd:token` dans l'espace de nommage commun DEVRAIT être utilisé, par exemple un renvoi.

[SD-56] Les schémas DEVRAIENT utiliser `xsd:key/xsd:keyref/xsd:unique` ou `xsd:ID/xsd:IDREF/xsd:IDREFS` pour identifier les contraintes de manière appropriée.

- [SD-57] Les schémas DEVRAIENT utiliser `xsd:ID/xsd:IDREF/xsd:IDREFS` pour un ou plusieurs renvois dans le document XML concerné pour identifier les contraintes de manière appropriée.
- [SD-62] Les schémas DEVRAIENT utiliser `xsd:key/xsd:unique/xsd:keyref` dans un domaine où l'unicité doit demeurer complètement.

### 3.5 Documentation du schéma

75. La spécification du schéma W3C permet à la documentation d'être incluse dans les schémas. Cela permet aux schémas de contenir des informations documentaires sur chaque structure. Chaque structure de schéma définie ou déclarée DOIT utiliser l'élément `xsd:documentation` pour la documentation requise et, le cas échéant, l'élément `xsd:appinfo` pour les informations de l'application dans l'élément `xsd:annotation`. L'élément `xsd:appinfo` communique le contexte et l'utilisation d'éléments dans les schémas à l'utilisateur qui n'a pas les mêmes antécédents ou compétences professionnelles que le développeur de schémas. En général, l'élément `xsd:documentation` est utilisé pour le matériel déchiffrable par l'homme et l'élément `xsd:appinfo` pour fournir des informations concernant les outils, les feuilles de style et d'autres applications.

#### 3.5.1 Documentation sur la structure du schéma

76. Les schémas XML peuvent également contenir des commentaires (à savoir `<!--comment -->`). Il est recommandé que les développeurs de schémas évitent d'utiliser cette technique, notamment si le but du commentaire est de décrire un aspect d'une structure de schéma donnée.

77. Il arrive que des commentaires du style HTML soient acceptables, comme lorsque le développeur veut insérer des délimiteurs visuels entre les structures comme des éléments et des types.

- [SD-58] Tous les schémas DEVRAIENT inclure une documentation sur leurs structures utilisant l'élément `xsd:documentation`. La documentation DEVRAIT décrire uniquement l'élément ou le type et NE DEVRAIT PAS contenir de détails sur l'application ou d'autres informations qui ne sont pas directement liées à la signification de la structure.
- [SD-59] Les commentaires (à savoir `<!--comment -->`) NE DEVRAIENT PAS être utilisés dans un schéma.
- [SD-60] La documentation NE DEVRAIT PAS remplacer les listes de codes utilisant une énumération.

#### 3.5.2 Documentation relative à l'en-tête du schéma

78. Tout comme la documentation sur la structure du schéma rend un schéma plus clair, la documentation relative à l'en-tête du schéma permet de concentrer l'information - but, utilisation et contenu d'un schéma - en un seul endroit à l'intérieur d'un schéma. La documentation relative à l'en-tête du schéma devrait être incluse dans tous les éléments au niveau du document et dans certaines composantes agrégées et de base telles que WIPOST3Code. Afin de faciliter le traitement de l'information dans l'en-tête par application, l'élément `xsd:appinfo` devrait être utilisé pour la documentation relative à l'en-tête du schéma.

79. Le tableau 1 donne la liste des éléments inclus dans l'en-tête des schémas.

Tableau 1. Informations sur la documentation relative à l'en-tête

Nom de l'en-tête	Description
Créé le	Date de création du schéma (obligatoire)
Date de la dernière modification	Date de la dernière modification du schéma (obligatoire)
Personne à contacter	Interlocuteur pour les questions sur le schéma (obligatoire)
URL des notes de diffusion	Une URL où les notes de diffusion du schéma sont publiées (facultatif)

L'exemple ci-après indique les éléments de schéma correspondants aux éléments de l'en-tête énumérés dans le tableau ci-dessus :

```
<xsd:annotation>
<xsd:appinfo> <com:SchemaCreatedDate>2011-05-28</com: SchemaCreatedDate>
  <com:SchemaLastModifiedDate>2020-06-01</com: SchemaLastModifiedDate>
  <com: SchemaContactPoint>xml.standards@wipo.int</com: SchemaContactPoint
>
  <com:
SchemaReleaseNoteURL>http://www.wipo.int/standards/XMLSchema/ST96/V4\_0/ReleaseNotes.pdf</com:SchemaReleaseNoteURL>
</xsd:appinfo></xsd:annotation>
```

80. La documentation relative à l'en-tête du schéma permet à un développeur de schéma de discerner facilement le but, l'utilisation et le contenu d'un schéma. Cette information est également très utile lorsqu'un développeur doit choisir un schéma à utiliser comme canevas pour la création d'un autre schéma.

[SD-61] Les schémas au niveau du document DOIVENT inclure une documentation relative à leur en-tête qui contient obligatoirement des renvois à la date de création, à la date de la dernière modification et à la personne à contacter ainsi qu'un renvoi facultatif à une URL où les notes de diffusion du schéma sont publiées.

#### 4. RÈGLES DE CONCEPTION DES INSTANCES

81. Les règles de conception des instances traitent des recommandations à suivre lorsque des données structurées sont fournies.

##### 4.1 Espaces de nommage dans les documents d'instance XML

###### 4.1.1 Validation des documents d'instance XML

82. Il est recommandé de valider un document d'instance XML en fonction d'un schéma. La validation d'un tel document garantit que son contenu répond à toutes les exigences du schéma qu'il valide. Le système de validation choisit le schéma fondé sur son emplacement exact comme précisé dans le document d'instance XML.

83. Les emplacements d'un schéma doivent revêtir la forme d'un URI. Ils sont normalement définis comme étant des URI fondés sur l'URL en raison des limites de résolvabilité des URI fondés sur l'URN.

84. L'emplacement d'un schéma peut figurer comme suit dans l'élément de base du document d'instance XML :

```
<?xml version="1.0" encoding="UTF-8"?>
<com:Contact xmlns:com="http://www.wipo.int/standards/XMLSchema/ST96/Common"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://www.wipo.int/standards/XMLSchema/ST96/Common
../V4_0/Common/Contact.xsd">
.....
</com:Contact>
```

85. L'attribut `schemaLocation` est une structure de schéma W3C qui associe un document d'instance XML à un schéma. Il n'est utilisé que lorsqu'un schéma a un espace de nommage cible. Dans l'exemple ci-dessus, l'identificateur d'espace de nommage "http://www.wipo.int/standards/XMLSchema/ST96/Common" est l'espace de nommage cible du schéma `Contact.xsd`.

[ID-01] Les instances XML DOIVENT être validées en fonction du schéma correspondant.

[ID-02] Les documents d'instance XML DOIVENT utiliser l'attribut `schemaLocation` pour associer l'espace de nommage cible à l'emplacement du fichier de schéma auquel l'instance XML se conforme.

[ID-03] Chaque déclaration d'attribut `xsi:schemaLocation` DEVRAIT contenir une URL facile d'accès.

#### 4.1.2 Déclaration et qualification de l'espace de nommage dans les documents d'instance XML

86. Comme dans le cas des schémas, un espace de nommage est indiqué dans l'élément de base d'un document d'instance XML à l'aide d'un identificateur d'espace de nommage avec un préfixe recommandé. Les documents d'instance XML doivent utiliser une qualification d'espace de nommage pour tous les éléments qui sont définis à l'annexe III de la norme ST.96 de l'OMPI.

87. Il convient de noter que :

- l'identificateur de l'espace de nommage dans un document d'instance XML doit être le même que l'identificateur de l'espace de nommage pour l'espace de nommage cible dans le schéma; et
- le préfixe de l'espace de nommage dans un document d'instance XML ne doit pas être le même que le préfixe de l'espace de nommage pour l'espace de nommage cible dans le schéma.

88. Les éléments et attributs dans les documents d'instance XML ne peuvent être qualifiés d'espaces de nommage que s'ils appartiennent à l'espace de nommage cible du schéma qui valide le document d'instance XML. C'est pourquoi tous les éléments et attributs globaux doivent être qualifiés d'espaces de nommage. Cependant, l'exigence relative aux éléments et attributs locaux qui appartiennent à l'espace de nommage cible du schéma dépend de l'établissement dans le schéma d'un mécanisme de commutation qui utilise les deux indicateurs suivants :

- `elementFormDefault`, et
- `attributeFormDefault`.

89. L'indicateur `elementFormDefault` contrôle la qualification d'espace de nommage des éléments locaux tandis que l'indicateur `attributeFormDefault` contrôle la qualification d'espace de nommage des attributs locaux. Ces deux indicateurs apparaissent comme des attributs de l'élément de base d'un schéma et ils peuvent avoir chacun la valeur `qualified` ou `unqualified` (implicite).

Par exemple,

```
<?xml version="1.0" encoding="UTF-8"?>
<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:pat="http://www.wipo.int/standards/XMLSchema/ST96/Patent"
targetNamespace="http://www.wipo.int/standards/XMLSchema/ST96/Patent"
elementFormDefault="qualified"
attributeFormDefault="qualified">
.....
</xsd:schema>
```

90. Ces déclarations exigent que tous les éléments et attributs locaux dans l'espace de nommage cible du schéma soient qualifiés d'espaces de nommage dans un document d'instance XML à moins que la valeur de l'attribut `form` ne soit altérée.

[ID-04] Chaque espace de nommage déclaré DEVRAIT recevoir un préfixe recommandé, à savoir le préfixe "com" pour toutes les structures de schéma à composante commune, "pat" pour toutes les structures de schéma à composante brevets, "tmk" pour toutes les structures de schéma à composante marques, "dgn" pour toutes les structures de schéma à composante dessins et modèles industriels, "tbl" pour toutes les structures de schéma à composante de tableau OASIS et "mathml" pour les structures de schéma à composante MathML.

[ID-05] Les documents d'instance XML NE DOIVENT PAS utiliser des espaces de nommage implicites.

#### 4.1.3 L'espace de nommage de l'instance du schéma W3C

91. La norme du schéma W3C a son propre espace de nommage appelé l'espace de nommage du schéma W3C, qui contient toutes les structures du schéma W3C utilisées dans les documents d'instance XML (`schemaLocation`, `noNamespaceSchemaLocation`, `type` et `nil`). Pour utiliser ces structures, l'espace de nommage du schéma W3C doit être déclaré dans l'élément de base d'un document d'instance XML à l'aide de l'identificateur de l'espace de nommage "`http://www.w3.org/2001/XMLSchema-instance`" :

```
<?xml version="1.0" encoding="UTF-8"?>
<com:Contact xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:com="http://www.wipo.int/standards/XMLSchema/ST96/Common">
...
</com:Contact>
```

92. Bien que défini par l'utilisateur, le préfixe `xsi` est le plus souvent visé dans la littérature et les références consacrées au schéma W3C comme le préfixe de l'espace de nommage pour les structures de ce schéma.

[ID-06] Les documents d'instance XML DEVRAIENT utiliser `xsi` comme un préfixe d'espace de nommage pour toutes les structures de schéma W3C.

#### 4.1.4 Portée de l'espace de nommage

93. Les espaces de nommage dans les documents d'instance XML ont un champ d'applicabilité dans un document d'instance XML. La portée d'un espace de nommage s'applique à l'élément déclaré (qui peut être l'élément de base) et à l'intégralité du contenu de cet élément. Un espace de nommage peut également être déclaré sur un élément autre que l'élément de base; c'est ce qu'on appelle une déclaration locale d'espace de nommage.

94. Bien que le traitement puisse être rendu plus efficace au moyen des déclarations locales d'espace de nommage, la capacité d'identifier visuellement tous les espaces de nommage déclarés dans un document d'instance en examinant l'élément de base est plus utile.

[ID-07] Les documents d'instance XML NE DOIVENT PAS utiliser des déclarations locales d'espace de nommage.

#### 4.2 Entités externes

95. Les images incrustées constituent une entité externe utilisée couramment, c'est-à-dire qu'un renvoi à un fichier d'image externe est inséré dans une instance de document à l'endroit où l'image devrait être montrée lorsque l'instance est établie. Elles sont le plus souvent des parties d'un document qui ne peuvent pas être codées et stockées à l'aide d'un ensemble de caractères. Les images incrustées PEUVENT être des dessins, des formules chimiques, des tableaux complexes et des caractères non définis.

96. Une instance XML peut renvoyer à d'autres fichiers XML fondés sur une DTD en tant qu'entité externe. Dans ce cas-là, elle ne peut pas être validée en fonction du schéma XML correspondant. Le fichier XML pourrait faire l'objet d'un renvoi au moyen de `xsd:anyURI` ou `xsd:string` pour indiquer l'emplacement du fichier externe de l'instance XML. Dans l'instance XML, cependant, il n'est pas recommandé de renvoyer à un fichier XML fondé sur une DTD.

97. Les images peuvent être incrustées dans une instance XML sous la forme d'images binaires incrustées codées dans un contenu de type `base64Binary` qui constitue le type de données incorporées du Consortium W3C XML ainsi que de renvois effectués à des fichiers d'image externes, c'est-à-dire des entités externes. Dans le présent document, cependant, les images doivent être désignées comme des entités externes car des images binaires incrustées peuvent inclure un code nuisible, par exemple des virus.

[ID-08] Les fichiers d'image externes DEVRAIENT être conformes à l'un des formats d'image recommandés dans la norme ST.96 de l'OMPI.

[ID-09] Les images DOIVENT être appelées fichiers externes.

[ID-10] Les listes de séquences DEVRAIENT suivre la norme ST.25 de l'OMPI.

**APPENDICE A – LISTE DES RÈGLES DE CONCEPTION**

*Règles de conception générales*

Règle ID	Règle
[GD-01]	Tous les schémas XML DOIVENT reposer sur des spécifications techniques du Consortium W3C qui ont obtenu le statut de recommandation.
[GD-02]	Les schémas DOIVENT se conformer : à la première partie du schéma XML : Structures ( <a href="http://www.w3.org/TR/xmlschema-1/">http://www.w3.org/TR/xmlschema-1/</a> ); et à la deuxième partie du schéma XML : Types de données ( <a href="http://www.w3.org/TR/xmlschema-2/">http://www.w3.org/TR/xmlschema-2/</a> ).
[GD-03]	Les schémas DOIVENT utiliser la norme ISO/IEC 10646 – UCS – Unicode pour l'ensemble des caractères. La norme UTF-8 DOIT être utilisée pour coder les caractères Unicode.
[GD-04]	Les noms de type, d'élément et d'attribut DOIVENT être composés de mots anglais, écrits selon l'orthographe anglaise figurant dans le dictionnaire Oxford English Dictionary. Les seules exceptions autorisées sont les sigles, les abréviations ou d'autres mots tronqués qui sont énumérés à l'appendice C.
[GD-05]	Les noms de type, d'élément et d'attribut DEVRAIENT comprendre uniquement des noms, des adjectifs et des verbes au présent à l'exception des sigles, abréviations ou autres mots tronqués qui sont énumérés à l'appendice C.
[GD-06]	Les caractères utilisés dans des noms de types d'élément et d'attribut DOIVENT être limités à la série suivante : 'a-z, A-Z and 0-9'.
[GD-07]	La longueur maximum d'un nom de composante DEVRAIT être de 35 caractères. L'utilisation d'abréviations ou d'acronymes est encouragée pour réduire la longueur d'un nom d'élément. Toute nouvelle abréviation ou tout nouvel acronyme DOIT être ajouté(e) à l'appendice C.
[GD-08]	Les noms de type, d'élément et d'attribut DEVRAIENT être concis et s'expliquer d'eux-mêmes.
[GD-09]	Les noms d'élément DOIVENT utiliser les caractères majuscules de type "camel" (UCC). Par exemple, <code>CountryCode</code> .
[GD-10]	Les noms de types DOIVENT utiliser la forme UCC et avoir un type suffixe. Par exemple, <code>ApplicantType</code> .
[GD-11]	Les noms d'attributs DOIVENT utiliser les caractères minuscules de type "camel" (LCC). Par exemple, <code>currencyCode="EUR"</code> .
[GD-12]	Les sigles et abréviations énumérés à l'appendice C DOIVENT toujours être utilisés en lieu et place du nom développé complet.
[GD-13]	Les sigles et abréviations au début d'une déclaration d'attribut DOIVENT tous figurer en minuscules. Par exemple, <code>biodepositDescriptionText</code> . Tous les autres sigles et abréviations utilisés dans une déclaration d'attribut DOIVENT figurer en majuscules comme indiqué dans l'appendice C.
[GD-14]	Les sigles et abréviations DOIVENT figurer comme indiqué dans l'appendice C pour les noms de type et d'élément.
[GD-15]	Les noms de type complexe DEVRAIENT inclure un terme de classe objet significatif.
[GD-16]	Lorsque nous avons besoin de deux termes de classe objet dans un nom de composante donnée compte tenu de la nature de l'entité opérationnelle, un type complexe d'association PEUT être utilisé. Les noms d'un type complexe d'association DEVRAIENT utiliser une structure de terme de classe objet du type complexe associant, un terme de propriété qui décrit la nature de l'association, et la classe objet du type complexe associé. Les termes de qualification peuvent précéder la classe objet et le terme de propriété. Par exemple, dans <code>ApplicantResidenceAddress</code> , <code>Applicant</code> est la classe objet de type complexe associant, <code>Residence</code> est la propriété et <code>Address</code> est la classe objet de type complexe associé.
[GD-17]	Les noms d'éléments basiques (ceux qui sont fondés sur un type simple ou complexe qui n'accepte pas les enfants) DEVRAIENT comporter un terme de classe objet, un terme de propriété et, le cas échéant, un terme de représentation. Les termes de qualification peuvent précéder le terme de classe objet et le terme de propriété.
[GD-18]	Un terme de classe objet DOIT toujours avoir la même signification sémantique dans un espace de nommage et PEUT comprendre plus d'un mot. Par exemple, <code>ContactInformation</code> .

Règle ID	Règle
[GD-19]	Le terme de propriété dans un nom DOIT être unique dans le contexte d'une classe objet mais PEUT être réutilisé dans différentes classes objets.
[GD-20]	Un terme de qualification PEUT être attaché à un terme de classe objet ou à un terme de propriété si cela s'avère nécessaire pour rendre un nom unique.
[GD-21]	Lorsqu'un nom comprend un terme de classe objet, un terme de propriété et un terme de représentation, le terme de classe objet DOIT précéder le terme de propriété et le terme de propriété DOIT précéder le terme de représentation. Un terme de qualification DEVRAIT précéder le terme associé de classe objet ou le terme de propriété.
[GD-22]	Si le terme de propriété finit avec le même mot que le terme de représentation (ou un mot équivalent), le terme de représentation DOIT être supprimé.
[GD-23]	Lorsqu'un terme de représentation est requis, les termes de représentation de l'appendice B DOIVENT être utilisés pour les termes de représentation dans les noms de composantes basiques.
[GD-24]	Dans un espace de nommage, tous les noms de type, d'élément et d'attribut DOIVENT être uniques.
[GD-25]	Le ou les mots dans un nom DEVRAIENT être utilisés au singulier sauf si le concept est lui-même au pluriel. Par exemple, <code>TotalMarkSeries</code> .
[GD-26]	Le nom d'un élément ou d'un type qui comprend une collection de composantes contextuellement reliées DEVRAIT avoir le suffixe "Bag". Par exemple, <code>EmailAddressBag</code> représente une collection d'éléments <code>EmailAddress</code> .
[GD-27]	Des mots qui relient comme "and", "of" et "the" NE DEVRAIENT PAS être utilisés dans les noms de type, d'élément et d'attribut à moins qu'ils ne fassent partie de la terminologie commerciale.
[GD-28]	Les noms de type, d'élément et d'attribut NE DOIVENT PAS être traduits, modifiés ou remplacés dans un but quel qu'il soit.
[GD-29]	Les noms de type et d'élément NE DOIVENT PAS faire référence à des numéros d'article et de règle. Par exemple, <code>PCTRule702C</code> pour le PCT.
[GD-30]	Les caractères utilisés dans les noms de fichier de schéma DOIVENT appartenir à la série suivante : 'a-z, A-Z, 0-9, caractère de soulignement "_", et point ".".
[GD-31]	Un nom de fichier de schéma DOIT comporter deux parties obligatoires avec un délimiteur et les informations facultatives de la version avec deux délimiteurs, c'est-à-dire : <nom de la composante>{"_" "V"<numéro de la version majeure>"_"<numéro de la version mineure>"}.<extension du fichier>. Par exemple, <code>EmailAddressType.xsd</code> , <code>ApplicationBody_V1_0.xsd</code> .
[GD-32]	Un nom de fichier du projet de schéma DOIT comporter quatre parties obligatoires avec deux délimiteurs et les informations facultatives de la version avec deux délimiteurs supplémentaires, c'est-à-dire : < nom de la composante>{"_" "V"<numéro de la version majeure>"_"<numéro de la version mineure>"}_"D"<numéro de la révision>".<extension du fichier>, par exemple, <code>Contact_D3.xsd</code> , <code>TrademarkApplication_V1_1_D1</code> . Si un projet de schéma est fondé sur un schéma existant et que son nom de fichier inclut les informations de la version, les numéros des versions majeure et mineure dans le nom de fichier du projet de schéma DEVRAIENT être ceux qui sont indiqués dans le fichier de schéma sur lequel est fondé le projet de schéma. Si le projet de schéma est nouveau, le numéro de la version majeure dans le nom de fichier du projet de schéma DEVRAIT être celui qui est indiqué dans l'espace de nommage correspondant et un numéro de version mineure dans le fichier du projet de schéma DEVRAIT être zéro "0".

#### Règles de conception d'un schéma

Règle ID	Règle
[SD-01]	Les modules du schéma de la composante brevets, de la composante marques, de la composante dessins et modèles industriels, de la composante indications géographiques et de la composante droit d'auteur DOIVENT utiliser la structure <code>xsd:import</code> pour renvoyer aux modules du schéma de la composante commune.
[SD-02]	Les modules du schéma de la composante commune NE DOIVENT PAS renvoyer aux modules du schéma de la composante brevets, de la composante marques, de la composante dessins et modèles industriels, de la composante indications géographiques ou de la composante droit d'auteur.

Règle ID	Règle
[SD-03]	Un module du schéma de la composante brevets, de la composante marques, de la composante dessins et modèles industriels, de la composante indications géographiques ou de la composante droit d'auteur PEUT renvoyer à des modules du schéma de la composante commune, à des schémas du module de la même composante et à des schémas approuvés de normes industrielles, mais NE DEVRAIT PAS renvoyer à des modules du schéma d'autres composantes, à une exception près : les schémas de la composante indications géographiques PEUVENT renvoyer à des schémas de la composante marques et inversement, car les secteurs d'activité en cause se recoupent partiellement.
[SD-04]	Les schémas existants DOIVENT être utilisés, le cas échéant, avant d'en créer de nouveaux.
[SD-05]	Les schémas DEVRAIENT utiliser dans toute la mesure du possible des éléments et types définis dans les schémas existants.
[SD-06]	Tous les types, éléments et attributs DOIVENT être déclarés globalement.
[SD-07]	Les schémas NE DOIVENT PAS utiliser la structure <code>xsd:redefine</code> .
[SD-08]	Les schémas DOIVENT utiliser des espaces de nommage.
[SD-09]	Les déclarations publiées des espaces de nommage NE DEVRAIENT PAS être modifiées.
[SD-10]	Les schémas DOIVENT déclarer l'espace de nommage du Consortium W3C.
[SD-11]	Les schémas DOIVENT utiliser les qualifications des espaces de nommage pour toutes les structures de schéma du Consortium W3C.
[SD-12]	Les schémas DOIVENT utiliser l'espace de nommage formaté en URI.
[SD-13]	Tous les schémas DOIVENT avoir des attributs <code>elementFormDefault</code> et <code>attributeFormDefault</code> avec la valeur "qualified" dans l'élément de base <code>xsd:schema</code> .
[SD-14]	Les noms des espaces de nommage DOIVENT avoir la structure suivante : <code>http://www.wipo.int/standards/XMLSchema/ST96/&lt;category&gt;</code> ; où <code>&lt;category&gt;</code> est un symbole identifiant le domaine du schéma utilisé : Common, Patent, Trademark, Design, Geographical Indication or Copyright.
[SD-15]	Les schémas DEVRAIENT utiliser "xsd" comme un préfixe d'espace de nommage pour toutes les structures de schéma du Consortium W3C, "com" pour tous les schémas de la composante commune, "pat" pour tous les schémas de la composante brevets, "tmk" pour tous les schémas de la composante marques, "dgn" pour tous les schémas de la composante dessins et modèles industriels, "gin" pour les schémas de la composante indications géographiques, "cpy" pour les schémas de la composante droit d'auteur, "tbl" pour les schémas de la composante OASIS Table et "mathml" pour les schémas de la composante MathML.
[SD-16]	Chaque schéma DOIT déclarer l'espace de nommage cible à l'aide de l'attribut <code>xsd:targetNamespace</code> .
[SD-17]	Le <code>targetNamespace</code> du schéma DOIT correspondre au nom de l'espace de nommage d'un des espaces de nommage déclarés mais pas à l'espace de nommage du Consortium W3C.
[SD-18]	Les schémas de la composante commune DOIVENT utiliser <code>http://www.wipo.int/standards/XMLSchema/ST96/Common</code> comme espace de nommage cible.
[SD-19]	Les schémas de la composante brevets DOIVENT utiliser <code>http://www.wipo.int/standards/XMLSchema/ST96/Patent</code> comme espace de nommage cible.
[SD-20]	Les schémas de la composante marques DOIVENT utiliser <code>http://www.wipo.int/standards/XMLSchema/ST96/Trademark</code> comme espace de nommage cible.
[SD-21]	Les schémas de la composante dessins et modèles industriels DOIVENT utiliser <code>http://www.wipo.int/standards/XMLSchema/ST96/Design</code> comme espace de nommage cible.
[SD-22]	Le schéma NE DEVRAIT PAS utiliser un espace de nommage implicite.
[SD-23]	Les nouvelles versions mineures d'un schéma DOIVENT pouvoir valider toutes les instances créées avec des versions mineures antérieures de ce schéma avec la même version majeure.
[SD-24]	La version majeure d'un schéma DOIT être renforcée si le nouveau schéma n'est pas en mesure de valider les instances de document XML existantes avec la version majeure utilisée.
[SD-25]	<i>Supprimé</i>
[SD-26]	<i>Supprimé</i>

Règle ID	Règle
[SD-27]	Le numéro de la version majeure ou mineure d'un schéma DEVRAIT être modifié lorsqu'un schéma inclus ou importé est actualisé.
[SD-28]	Le numéro de la version majeure ou mineure d'un schéma DEVRAIT être modifié lorsqu'un schéma inclus ou importé est actualisé.
[SD-29]	Dans la création d'un nouveau schéma, les versions les plus récentes de tous les schémas inclus ou importés DEVRAIENT être utilisées.
[SD-30]	L'attribut <code>version</code> du schéma W3C DOIT comprendre à la fois le numéro de la version majeure et le numéro de la version mineure pour chaque fichier de schéma dans le format suivant : "V"<major version number>"_ "<minor version number>.
[SD-31]	Le schéma pour une composante document DOIT déclarer un attribut nécessaire appelé <code>st96Version</code> dans l'élément de base avec une valeur fixe correspondant à la version de diffusion du schéma XML de la norme ST.96, dans le format suivant : "V"<major version number>"_ "<minor version number>.
[SD-32]	Les schémas ST.96 DEVRAIENT être élaborés avec le même niveau de cardinalité et de granularité afin de faciliter les possibilités de conversion des instances ST.96 avec les instances ST.36, ST.66 ou ST.86.
[SD-33]	Les énumérations contenues dans les schémas NE DEVRAIENT PAS inclure les valeurs "Autre" ou "inconnue" à moins qu'elles ne soient nécessaires.
[SD-34]	MathML, version 3, DEVRAIT être utilisée pour les formules mathématiques.
[SD-35]	OASISTable.xsd, qui repose sur le schéma de tableau d'échange XML OASIS, DEVRAIT être utilisé pour les tableaux.
[SD-36]	Les schémas de normes industrielles DEVRAIENT être incorporés par renvoi uniquement et après l'approbation préalable du Comité des normes de l'OMPI.
[SD-37]	Les schémas DEVRAIENT dans toute la mesure du possible utiliser des types simples.
[SD-38]	Des listes de codes DEVRAIENT être indiquées comme des listes d'énumération de type simple.
[SD-39]	Une liste de codes PEUT être indiquée comme une union de types simples.
[SD-40]	La norme ST.3 de l'OMPI (codes à deux lettres) DOIT être utilisée pour la représentation des offices de propriété intellectuelle ainsi que pour les pays/organisations désignés et prioritaires. Par exemple, <code>PriorityCountryCode="EP"</code> .
[SD-41]	La norme ISO 3166-1-Alpha-2 Éléments de code (codes de pays à deux lettres) DOIT être utilisée pour la représentation des noms de pays, des dépendances et autres zones d'intérêt géopolitique particulier, sur la base des listes de noms de pays provenant des Nations Unies.
[SD-42]	La norme ISO 639-1 (codes de langues à deux lettres) DOIT être utilisée pour les codes des langues.
[SD-43]	Les schémas DOIVENT indiquer les éléments et les attributs pour les valeurs de date et d'heure en utilisant les types de données relatives à la date et à l'heure du schéma XML du W3C.
[SD-44]	La norme ISO 4217-Alpha (codes pour la représentation des monnaies à trois lettres) DOIT être utilisée pour les codes des monnaies.
[SD-45]	Les caractères utilisés dans les énumérations DOIVENT se limiter aux caractères suivants : 'a-z, A-Z, 0-9, espace " " et caractère de soulignement "_ '.
[SD-46]	Les valeurs d'énumération DEVRAIENT être suffisantes d'un point de vue sémantique et en anglais, et utiliser un nombre minimum de caractères. Les valeurs DEVRAIENT être tirées du langage utilisé couramment dans les activités relatives à la propriété intellectuelle.
[SD-47]	Des types complexes abstraits PEUVENT être utilisés.
[SD-48]	Les schémas DEVRAIENT uniquement utiliser des attributs pour définir des données non commerciales. Par exemple, il est autorisé d'utiliser un attribut intitulé <code>sequenceNumber</code> pour les numéros d'ordre.
[SD-49]	Un indicateur d'occurrence NE DEVRAIT PAS être utilisé pour préciser une restriction qui est l'élément défaut dans un schéma. Par exemple, <code>minOccurs="1"</code> et <code>maxOccurs="1"</code> NE DEVRAIENT PAS être utilisés mais <code>minOccurs="2"</code> et <code>maxOccurs="3"</code> sont acceptables.
[SD-50]	Les éléments vides NE DOIVENT PAS être définis dans les schémas, sauf pour le saut de ligne.
[SD-51]	L'attribut <code>use</code> dans une utilisation d'attribut DEVRAIT être omis lorsque l'attribut visé est facultatif car l'élément implicite est <code>use="optional"</code> .
[SD-52]	Les schémas NE DEVRAIENT PAS utiliser le compositeur <code>all</code> .

Règle ID	Règle
[SD-53]	Les schémas PEUVENT utiliser des indicateurs d'occurrence dans les compositeurs <code>xsd:sequence</code> et <code>xsd:choice</code> .
[SD-54]	Un schéma NE DOIT PAS contenir un type complexe avec <code>xsd:any</code> pour l'extensibilité.
[SD-55]	Les schémas NE DOIVENT PAS utiliser les groupes de substitution.
[SD-56]	Les schémas DEVRAIENT utiliser <code>xsd:key/xsd:keyref/xsd:unique</code> ou <code>xsd:ID/xsd:IDREF/xsd:IDREFS</code> pour identifier les contraintes de manière appropriée.
[SD-57]	Les schémas DEVRAIENT utiliser <code>xsd:ID/xsd:IDREF/xsd:IDREFS</code> pour un ou plusieurs renvois dans le document XML concerné pour identifier les contraintes de manière appropriée.
[SD-58]	Tous les schémas DEVRAIENT inclure une documentation sur leurs structures utilisant l'élément <code>xsd:documentation</code> . La documentation DEVRAIT décrire uniquement l'élément ou le type et NE DEVRAIT PAS contenir de détails sur l'application ou d'autres informations qui ne sont pas directement liées à la signification de la structure.
[SD-59]	Les commentaires (c'est-à-dire <code>&lt;&lt;!--comment --&gt;</code> ) NE DEVRAIENT PAS être utilisés dans un schéma.
[SD-60]	La documentation NE DEVRAIT PAS remplacer les listes de codes utilisant une énumération.
[SD-61]	Les schémas au niveau du document DOIVENT inclure une documentation relative à leur en-tête qui contient obligatoirement des renvois à la date de création, à la date de la dernière modification et à la personne à contacter ainsi qu'un renvoi facultatif à une URL où les notes de diffusion du schéma sont publiées.
[SD-62]	Les schémas DEVRAIENT utiliser <code>xsd:key/xsd:unique/xsd:keyref</code> dans un domaine où l'unicité doit demeurer complètement.
[SD-63]	Le schéma pour une composante document DOIT déclarer un attribut facultatif appelé <code>ipoVersion</code> dans l'élément de base avec une valeur fixe correspondant à la version de diffusion du schéma d'application de l'office de propriété intellectuelle, dans le format suivant : <code>&lt;ST.3 Code&gt;"_ "V"&lt;major version number&gt;"_ "&lt;minor version number&gt;</code> , par exemple, "US_V2_0"..
[SD-64]	Dossier de diffusion de schéma, un nom de fichier de schéma au niveau d'un document et un nom de fichier de schéma mis à plat DOIVENT comprendre des informations correspondant à la version y compris le numéro de la version majeure et celui de la version mineure.
[SD-65]	Les schémas de la composante indications géographiques DOIVENT utiliser <a href="http://www.wipo.int/standards/XMLSchema/ST96/GeographicalIndication">http://www.wipo.int/standards/XMLSchema/ST96/GeographicalIndication</a> comme espace de nommage cible.
[SD-66]	Les schémas de la composante droit d'auteur DOIVENT utiliser <a href="http://www.wipo.int/standards/XMLSchema/ST96/Copyright">http://www.wipo.int/standards/XMLSchema/ST96/Copyright</a> comme espace de nommage cible.

#### Règles de conception de l'instance

Règle ID	Règle
[ID-01]	Les instances XML DOIVENT être validées en fonction du schéma correspondant.
[ID-02]	Les documents d'instance XML DOIVENT utiliser l'attribut <code>schemaLocation</code> pour associer l'espace de nommage cible à l'emplacement du fichier du schéma auquel l'instance XML se conforme.
[ID-03]	Chaque déclaration d'attribut <code>xsi:schemaLocation</code> DEVRAIT contenir une URL facile d'accès.
[ID-04]	Chaque espace de nommage déclaré DEVRAIT recevoir un préfixe recommandé, à savoir le préfixe "com" pour toutes les structures de schéma à composante commune, "pat" pour toutes les structures de schéma à composante brevets, "tmk" pour toutes les structures de schéma à composante marques, "dgn" pour toutes les structures de schéma à composante dessins et modèles industriels, "tbl" pour toutes les structures de schéma à composante de tableau OASIS et "mathml" pour les structures de schéma à composante MathML.
[ID-05]	Les documents d'instance XML NE DOIVENT PAS utiliser des espaces de nommage implicites.
[ID-06]	Les documents d'instance XML DEVRAIENT utiliser <code>xsi</code> comme un préfixe d'espace de nommage pour toutes les structures de schéma W3C.

<b>Règle ID</b>	<b>Règle</b>
[ID-07]	Les documents d'instance XML NE DOIVENT PAS utiliser des déclarations locales d'espace de nommage.
[ID-08]	Les fichiers d'image externes DEVRAIENT être conformes à l'un des formats d'image recommandés dans la norme ST.96 de l'OMPI.
[ID-09]	Les images DOIVENT être appelées fichiers externes.
[ID-10]	Les listages de séquences DEVRAIENT suivre la norme ST.25 de l'OMPI.

**APPENDICE B – TERMES DE REPRÉSENTATION**

Terme	Définition	Data Type
Amount (Montant)	Une valeur monétaire	AmountType
Category (Catégorie)	Une division ou un sous-ensemble spécifiquement défini d'un système de classification dans lequel tous les éléments partagent le même concept de taxonomie.	xsd:token
Code	Une combinaison d'un ou plusieurs chiffres, lettres ou caractères spéciaux, qui est utilisée pour une signification spéciale. Représente des valeurs limites, déterminées à l'avance, ou en format libre.	xsd:token
Date	La notion d'un moment spécifique, désignée par l'année, le mois et le jour	DateType ou xsd:date
Identifiant (Identificateur)	Une combinaison d'un ou plusieurs nombres entiers, lettres, caractères spéciaux qui identifie de manière unique une instance spécifique d'un objet commercial mais qui peut ne pas avoir une signification facile à définir.	xsd:token ou IdentifierType
Indicator (Indicateur)	Un signal de la présence, de l'absence ou de l'exigence de quelque chose. Les valeurs recommandées sont "Vrai", "Faux", "1" ou "0" et ne doivent pas être vides. Ces valeurs sont sensibles à la casse.	xsd:boolean
Measure (Mesure)	Une mesure est une valeur numérique déterminée en mesurant un objet avec l'unité de mesure donnée. MeasureType est utilisé pour représenter un type de dimension physique comme la température, la longueur, la vitesse, la largeur, le poids, le volume et la latitude d'un objet. En termes plus précis, MeasureType devrait être utilisé pour mesurer les propriétés intrinsèques ou physiques d'un objet considéré comme un tout.	MeasureType
Name (Nom)	La désignation d'un objet exprimé en un mot ou en une phrase.	xsd:string
Number (Nombre)	Une série de chiffres ou de caractères alphanumériques désignant l'étiquette, la valeur, la quantité ou l'identification.	xsd:positiveInteger, xsd:string ou xsd:token
Percent (Pour cent)	Un chiffre qui représente la partie d'un tout qui sera divisé par 100.	xsd:decimal
Quantity (Quantité)	Une quantité est un nombre compté d'unités non monétaires, y compris éventuellement des fractions. Quantity est utilisé pour représenter un nombre compté de choses. Quantity devrait être utilisé pour des propriétés simples d'un objet considéré comme un composite, une collection ou un conteneur afin de quantifier ou de compter ses éléments. Quantity devrait toujours exprimer un nombre compté de choses et la propriété sera totale, expédiée, chargée, stockée. QuantityType devrait être utilisé pour les composantes nécessitant des informations sur les unités; et xsd:nonNegativeInteger devrait être utilisé pour les composantes comptables qui ne nécessitent pas d'informations sur les unités.	QuantityType ou xsd:nonNegativeInteger
Rate (Taux)	Une quantité ou un montant mesuré en fonction d'une autre quantité ou d'un autre montant.	xsd:decimal
Text (Texte)	Une chaîne de caractères formatés ou non formatés, généralement sous la forme de mots (comprend les abréviations et commentaires).	xsd:string, LocalizedTextType, OrderedTextType ou PhraseType
Time (Temps)	La désignation d'un moment chronologique précis dans une période.	xsd:time
DateTime (Date et heure)	La date et l'heure d'un événement.	xsd:datetime
URI	L'Uniform Resource Identifier qui identifie l'endroit où se trouve le fichier.	xsd:anyURI

## APPENDICE C – LISTE DES SIGLES ET ABRÉVIATIONS

Pour les abréviations, en principe, le premier caractère du mot ou de la phrase devrait être en majuscules et les derniers caractères devraient être en minuscules (par exemple, "Pre", "BioDeposit"), sauf pour les abréviations qui sont déjà connues, comme "IDREF", "W3C", "ST3". En ce qui concerne les sigles, tous les caractères devraient être en majuscules, par exemple "OMPI". Les acronymes et abréviations ci-après NE DEVRAIENT PAS être considérés dans le contexte des codes de langue, de monnaie, d'office ou de pays, qui sont énumérés dans la norme ST.96 de l'OMPI, car ils pourraient produire des doublons. Ces codes sont fondés sur les codes de langue de la norme ISO 639-1, les codes de monnaie de la norme ISO 4217, les codes de la norme ST.3 de l'OMPI et les codes de pays de la norme ISO 3166-1, respectivement.

AT	Fichier d'autorité
Alt	Texte alternatif pour une image
B	Caractère gras
BioDeposit	Dépôt biologique
Br	Saut de ligne
CDX	Format de fichier CambridgeSoft proprietary ChemDraw
CPC	Classification coopérative des brevets
DD	Description de la définition
Del	Texte supprimé
DL	Liste des définitions
DOI	Identificateur d'objet numérique
DT	Terme de définition
DTD	Définition de type de document
DWF	Format de conception sur le Web
DWG	Dessin
ECLA	Classification européenne
EIDR	Registre d'identification des loisirs
ExtRef	Renvois externes au document XML concerné
GI	Indication géographique
H<n>	"n" indique le niveau de l'en-tête avec une valeur spécifique constituée d'un nombre entre 1 et 15. Cela signifie que, dans la valeur d'énumération, cette abréviation indique une valeur entre H1 et H15. Par exemple, "H1" signifie "En-tête 1".
I	Italique
IB	Bureau international
ID	Identificateur pour l'identification des systèmes
IDREF	Renvoi à l'identificateur
IDREFS	Renvois à l'identificateur
IGES	Initial Graphic Exchange Specification
IGO	Organisation intergouvernementale dont la constitution prévoit la protection des droits de propriété intellectuelle ou qui agit dans le cadre d'un traité relatif à la propriété intellectuelle
INID	Codes d'identification numérique internationalement agréés en matière de données bibliographiques
Ins	Texte inséré
IP	Propriété intellectuelle
IPC	Classification internationale des brevets
IPCR	Réforme de la classification internationale des brevets

IPO	Office de propriété intellectuelle
IPR	Droit de propriété intellectuelle
ISMN	Numéro international de la musique
ISNI	Code international normalisé des noms
ISO	Organisation internationale de normalisation
LCC	Caractères bas de casse de type "camel"
LI	Liste
LOR	Licence de droit
MPEG	Moving Picture Experts Group
MOL	Format de fichier pour disposer d'informations sur les atomes, les liens, la connectivité et les coordonnées d'une molécule
NB	Format de fichier pour les blocs-notes Mathematica
NPL	Littérature non-brevet
NUTS	Nomenclature des unités territoriales statistiques
O	Over score
OASIS	Organization for the Advancement of Structured Information Standards
OCR	Reconnaissance optique des caractères
OL	Liste par ordre
P	Paragraphe
PAN	Numéro de compte principal
PCT	Traité de coopération en matière de brevets
PKCS7	En <a href="#">cryptographie</a> , PKCS est un ensemble de normes de <a href="#">cryptographie à clé publique</a> et <a href="#">PKCS #7</a> (PKCS7) désigne la norme de syntaxe de message cryptographique qui décrit la syntaxe générale pour les données auxquelles la cryptographie peut s'appliquer, comme les signatures et les enveloppes numériques.
Pre	Texte préformaté
S	Texte biffé
SEQL	Listage de séquence
SOC	Code de société
SPC	Certificat complémentaire de protection
ST3	Norme ST.3 de l'OMPI
ST13	Norme ST.13 de l'OMPI
Sub	Indice
Sup	Exposant
SVG	Image au format Scalable Vector Graphics
SWF	Small Web Format
SWIFT	Society for Worldwide Interbank Financial Telecommunication
ThreeDM	Modélisation en trois dimensions
ThreeDS	3D Studio
TISA	Code alphanumérique du système d'information sur les territoires de la CISAC
TISN	Code numérique du système d'information sur les territoires de la CISAC
TSG	Spécialités traditionnelles garanties

U	Soulignement
UCC	Caractères haut de casse de type "camel"
UL	Liste sans ordre
UPOV	Union internationale pour la protection des obtentions végétales
URI	Uniform Resource Identifier
URL	Uniform Resource Locator
URN	Uniform Resource Name
W3C	World Wide Web Consortium
WIPO	Organisation Mondiale de la Propriété Intellectuelle
WMV	Windows Media Video
XML	eXtensible Markup Language
XSD	Description de format de document XML (W3C)

## APPENDICE D – RÉFÉRENCES

### *Normes de l'OMPI*

- Norme ST.3 de l'OMPI : Codes à deux lettres pour la représentation des États, autres entités et organisations intergouvernementales;
- Norme ST.16 de l'OMPI : Identification de différents types de documents de brevet
- Norme ST.25 de l'OMPI : Présentation des listages des séquences de nucléotides et d'acides aminés
- Norme ST.26 de l'OMPI : Présentation des listages des séquences de nucléotides et d'acides aminés en langage XML
- Norme ST.27 de l'OMPI : Échange de données sur la situation juridique des brevets
- Norme ST.36 de l'OMPI : Utilisation du XML dans le traitement de l'information en matière de brevets
- Norme ST.37 de l'OMPI : Fichier d'autorité des documents de brevet publiés
- Norme ST.60 de l'OMPI : Données bibliographiques relatives aux marques
- Norme ST.61 de l'OMPI : Recommandation concernant l'échange de données sur la situation juridique des marques
- Norme ST.66 de l'OMPI : Utilisation du XML dans le traitement de l'information en matière de marques
- Norme ST.67 de l'OMPI : Gestion électronique des éléments figuratifs des marques
- Norme ST.80 de l'OMPI : Données bibliographiques relatives aux dessins et modèles industriels
- Norme ST.86 de l'OMPI : Utilisation du langage XML dans le traitement de l'information en matière de dessins et modèles industriels
- Norme ST.87 de l'OMPI : Échange de données sur la situation juridique des dessins et modèles industriels

### *Normes industrielles*

- Langage de définition de schéma XML du W3C (XSD), série de recommandations—Schéma XML, Partie 1 : Structures et schéma XML; Partie 2 : Types
- Appel à observations 2119 émis par l'Internet Engineering Task Force
- Norme ISO/IEC 10646- Technologies de l'information – Jeu universel de caractères codés sur plusieurs octets (JUC)
- Norme ISO 11179 – Technologies de l'information – Registres de métadonnées (RM) – Partie 5 : Principes de dénomination et d'identification
- Norme ISO 3166-1:2006 – Codes pour la représentation des noms de pays et de leurs subdivisions – Partie 1 : Codes de pays
- Norme ISO 639-1:2002 – Codes pour la représentation des noms de langue – Première partie : code Alpha2
- Norme ISO 8601:2004 – Éléments de données et formats d'échange – Échange d'information – Représentation de la date et de l'heure
- Norme ISO 4217:2008 – Codes pour la représentation des monnaies et des types de fonds
- Schéma de tableau XML OASIS : <http://www.oasis-open.org/docbook/xmlschema/1.0b1/calstbl.xsd>
- MathLab : MathML, version 3. Voir <http://www.w3.org/TR/MathML3> pour une description complète.

[L'annexe II suit]