

NORME ST.96 - ANNEXE I

RÈGLES ET CONVENTIONS DE CONCEPTION XML

Version 1.0

adoptée par le Comité des normes de l'OMPI (CWS) à sa deuxième session le 4 mai 2012

Table des matières

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	5
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	10
2.5.1 Déclaration et qualification des espaces de nommage.....	10
2.5.2 Espaces de nommage dans le schéma XML.....	10
2.5.3 Espaces de nommage cibles.....	11
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 des schémas	13
2.6.2.1 Espace de nommage dans le contrôle des schémas	13
2.6.2.2 Conventions de nommage des fichiers dans le contrôle des schémas	13
2.6.2.3 Attribut incorporé de la version du schéma XML dans le contrôle des schémas.....	14
2.6.2.4 Attribut <code>schemaVersion</code> défini par l'utilisateur dans le contrôle des schémas pour les documents XML.....	14
2.7 Possibilités de conversion avec d'autres normes XML de l'OMPI	14
2.8 Schémas de normes industrielles.....	15
3. CONVENTIONS DE CONSTRUCTION DE SCHÉMAS XML.....	15
3.1 Définitions des types.....	15
3.1.1 Types simples.....	15
3.1.1.1 Types de données incorporées du W3C.....	16
3.1.1.2 Types de données définis par l'utilisateur	16
3.1.2 Types complexes.....	16

Annexe I, page 2

3.2	Éléments et attributs	16
3.2.1	Éléments ou attributs	17
3.2.2	Éléments	17
3.2.2.1	Cardinalité des éléments	17
3.2.2.2	Éléments vides	17
3.2.3	Attributs	17
3.2.4	Regroupement des éléments et attributs	17
3.3	Extension et restriction	18
3.3.1	Extension	18
3.3.2	Restriction	18
3.3.3	Groupes de substitution	18
3.4	Contraintes en matière d'identité	18
3.5	Documentation du schéma	19
3.5.1	Documentation relative à l'en-tête du schéma	19
4.	RÈGLES DE CONCEPTION DES DOCUMENTS	20
4.1	Espaces de nommage dans les documents XML	20
4.1.1	Validation des documents XML	20
4.1.2	Déclaration et qualification de l'espace de nommage dans les documents XML	20
4.1.3	L'espace de nommage du schéma W3C	21
4.1.4	Portée de l'espace de nommage	21
4.2	Entités externes	22
	APPENDICE A – RÉSUMÉ DES RÈGLES DE CONCEPTION	23
	Règles de conception générales	23
	Règles de conception d'un schéma	24
	APPENDICE B – TERMES DE REPRÉSENTATION	28
	APPENDICE C – LISTE DES SIGLES ET ABRÉVIATIONS	29
	APPENDICE D – RÉFÉRENCES	30
	Normes de l'OMPI	30
	Normes industrielles	30

Annexe I, page 3

1. INTRODUCTION

1.1 *Aperçu*

1. Les règles et conventions de conception XML donnent l'occasion de promouvoir l'harmonie entre les trois catégories de titres de propriété industrielle et de faciliter l'échange de données entre les offices de propriété industrielle en utilisant des schémas XML qui peuvent être réutilisés et qui sont interoperables.

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 documents XML concernant 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 industrielle.

1.3 *Comment utiliser le présent document*

3. Le présent document a pour but d'être utilisé par les offices de propriété industrielle, les fournisseurs de données de propriété industrielle et la communauté élargie de la propriété industrielle. La communauté de la propriété industrielle 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 industrielle peuvent s'inspirer de ce document pour élaborer leurs propres règles de conception internes. Le présent document devrait être consulté 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 ce document;
- La Section 2, Conventions de conception XML, définit les règles de haut niveau qui s'appliquent aux activités d'élaboration de schémas et de documents;
- La Section 3, Conventions de construction de schémas XML, définit des règles de conception spécifiques servant à utiliser les spécifications du schéma W3C pour la création de schémas XML; et
- La Section 4, Règles de conception des documents, définit des règles de conception spécifiques pour créer des documents.

5. En outre, ce guide contient quatre appendices :

- L'appendice A, Résumé des règles de conception, résume les règles de conception trouvées 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.

Annexe I, page 4

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;
- 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 W3C —Première partie du schéma XML : Structures et schémas XML; Deuxième partie 2 : Type de données;
- le terme schéma compatible avec la norme ST.96 désigne un schéma compatible avec les éléments du schéma de la norme ST.96 et les règles et conventions de conception XML pour la propriété industrielle, c'est-à-dire l'annexe I de la norme ST.96;
- le terme schéma conforme à la norme ST.96 désigne un schéma qui n'a pas été étendu et qui soutient des contraintes exprimées par un schéma ST.96.

7. Dans le présent document,

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

8. Ces règles et conventions de conception contiennent certains mots clés qui ont une signification explicite. Ces mots clés, qui sont fondés sur les définitions figurant dans l'appel à observations 2119 émis par l'Internet Engineering Task Force, sont les suivants (les formes en minuscules de ces mots sont utilisées au sens anglais habituel);

- **DOIT** ou **DOIVENT** : Ce mot, ou les termes **REQUIS** ou **DEVRA** ou **DEVront**, signifie que la définition est une condition absolue de la spécification;
- **NE DOIT PAS** ou **NE DOIVENT PAS** : Cette expression, ou l'expression **NE DEVRA PAS** ou **NE DEVront PAS**, signifie que la définition est une interdiction absolue de la spécification;
- **DEVRAIT** ou **DEVRAIENT** : Ce mot, ou l'adjectif **RECOMMANDÉ(S)**, signifie qu'il peut y avoir des raisons valables dans des circonstances particulières pour ignorer un terme particulier mais toutes les conséquences doivent être comprises et soigneusement évaluées avant de choisir une trajectoire différente;
- **NE DEVRAIT PAS** ou **NE DEVRAIENT PAS** : Ces termes, ou les termes **NON RECOMMANDÉ(S)**, signifient qu'il peut y avoir des raisons valables dans des circonstances particulières lorsque le comportement particulier est acceptable ou même utile mais toutes les conséquences doivent être comprises et le cas soigneusement évalué avant d'appliquer un comportement assorti de ce label; et
- **PEUT** ou **PEUVENT** : Ce mot ou l'adjectif **FACULTATIF(S)** signifie qu'un élément est véritablement facultatif. Une mise en œuvre qui ne comporte pas une option particulière **DOIT** être prête à permettre l'interopérabilité avec une autre mise en œuvre qui comporte l'option encore que, peut-être, avec une fonctionnalité réduite. Dans le même ordre d'idées, une mise en œuvre qui comporte une option particulière **DOIT** être prête à permettre l'interopérabilité avec une autre mise en œuvre qui ne comporte pas l'option (sauf bien sûr pour la caractéristique que l'option fournit).

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 la mise en œuvre, sera remplacé par une valeur d'instance spécifique.
- **" "** : indique que le texte entre guillemets doit être utilisé *in extenso* dans la mise en œuvre.
- **{ }** : indique que la mise en œuvre est facultative.
- **Courier** : indique les mots clés XML, les noms de balise et les codes XML, en caractères **Courier**.

Annexe I, page 5

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 documents.
- b) La valeur "*nn*" définit le numéro d'ordre de la règle.

Par exemple, l'identificateur [GD-10] identifie la dixième règle de conception générale.

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 l'assise commune de l'élaboration de données et documents aux fins de l'information sur la propriété industrielle.

- [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] La norme ISO/IEC 10646 – UCS – Unicode DOIT être utilisée pour l'ensemble des caractères. 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. Elles se prêtent également au nommage des fichiers.

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 un regroupement ou une agrégation de données logique (dans un modèle de données logique) auquel un bien 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 en 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 un mot ou des mots qui aident à définir et distinguer un élément de donnée de ses autres éléments de donnée connexes et qui peuvent être associés au terme de classe objet ou de propriété si cela s'avère nécessaire pour rendre un nom unique.

Annexe I, page 6

- [GD-04] Les noms de types, d'éléments et d'attributs DOIVENT être composés de mots anglais, écrits selon l'orthographe anglaise figurant dans le dictionnaire Oxford English Dictionary, y compris les noms propres à un office, à l'exception des sigles, des abréviations ou d'autres mots tronqués qui sont énumérés dans l'appendice C.
- [GD-05] Les noms de types, d'éléments et d'attributs DOIVENT uniquement contenir des noms, des adjectifs et des verbes au présent.
- [GD-06] Les caractères utilisés dans des noms de types d'éléments et d'attributs DOIVENT être limités à la série suivante : {a-z, A-Z et 0-9}.
- [GD-07] La longueur maximale d'un nom DEVRAIT être de 35 caractères.
- [GD-08] Les noms de types, d'éléments et d'attributs DEVRAIENT être concis et s'expliquer d'eux-mêmes.
- [GD-09] Les noms d'éléments DOIVENT figurer en caractères majuscules de type "camel" (UCC). Par exemple, `CountryCode`.
- [GD-10] Les noms de types DOIVENT se présenter sous la forme UCC + type suffixe. Par exemple, `ApplicantType`.
- [GD-11] Les noms d'attributs DOIVENT figurer en caractères minuscules de type "camel" (LCC). Par exemple, `currencyCode="EUR"`.
- [GD-12] Les sigles et les abréviations indiqués dans l'appendice C DOIVENT toujours être utilisés en lieu et place du nom développé complet.
- [GD-13] Les sigles et les abréviations au début d'une déclaration d'attribut DOIVENT figurer en caractères minuscules. Tous les autres sigles et abréviations utilisés dans une déclaration d'attribut DOIVENT figurer en caractères majuscules.
- [GD-14] Les sigles et abréviations DOIVENT figurer en caractères majuscules pour tous les noms de type et d'élément.
- [GD-15] Les noms de type complexe DEVRAIENT inclure un terme de classe objet significatif.
- [GD-16] Les noms de type complexe d'association DEVRAIENT utiliser une structure de classe objet du type complexe associant (nature de l'association), et la classe objet du type complexe associé et des qualificatifs. Par exemple, `ApplicantResidenceAddress` : `Applicant` est le postulant d'une classe objet de type complexe associant, `Residence` est la propriété et `Address` est la classe objet de type complexe associé.
- [GD-17] Les noms de type simple et d'élément atomique (s'il n'a pas d'enfants) DEVRAIENT comporter le terme de classe objet, le terme de propriété, le terme de représentation et le terme de qualificatif.
- [GD-18] Un terme de classe objet DOIT toujours avoir la même signification sémantique dans un espace de nommage et PEUT avoir 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] Le terme de classe objet DOIT occuper dans le nom la première place (la plus à gauche), le terme de propriété la place suivante et le terme de représentation la dernière place (la plus à droite). Le 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 alors être supprimé.
- [GD-23] Les termes de représentation à l'appendice B DOIVENT être utilisés pour les termes de représentation dans les noms de composante.
- [GD-24] Dans un espace de nommage, tous les noms de type, d'élément et d'attribut (qu'ils soient déclarés localement ou globalement) DOIVENT être uniques.

Annexe I, page 7

- [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'une collection DEVRAIT utiliser le suffixe "Bag". Par exemple, `EmailAddressBag` représente une collection d'`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. Par exemple, `GoodsAndServices`.
- [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 fichiers d'un schéma et les noms d'un schéma sont souvent associés. Les noms de fichiers d'un schéma reposent sur les noms d'un 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 d'un schéma sont liées aux règles des conventions de nommage XML dans le présent document.

15. Les conventions de nommage des fichiers d'un schéma constituent une partie importante de la stratégie d'établissement des versions d'un schéma. Les noms de fichier d'un schéma doivent être utilisés systématiquement pour veiller à ce que les utilisateurs puissent faire la différence entre les versions d'un schéma. Par conséquent, les règles dans la présente section sont étroitement liées aux règles de la section `Schema Versioning` du présent document.

16. Le fichier d'un schéma devrait avoir les informations de la version. Il est recommandé d'inclure l'identificateur de la version dans le nom de fichier du schéma. Dans le cas d'un schéma qui en est au stade de projet (projet de schéma), des révisions sont autorisées. Les projets de schéma doivent être indiqués comme tels dans le nom de fichier du schéma, avec la lettre "D" et le numéro de la révision.

- [GD-30] Les caractères utilisés dans les noms de fichier d'un schéma DOIVENT être limités à la série suivante : {a-z, A-Z, 0-9, tiret (-), et point (.)}.
- [GD-31] Le nom de fichier d'un schéma DOIT comporter cinq parties avec trois 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-V1-1.xsd`, `languageCode-V1-0.xsd`, `id-V1-2.xsd`.
- [GD-32] Le nom de fichier du projet de schéma DOIT comporter sept parties avec quatre délimiteurs, 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-V1-0-D3.xsd`.

2.3 Stratégie de modularité

17. La norme ST.96 de l'OMPI repose pour beaucoup sur la modularité dans la conception du schéma. La division de la réalisation physique des schémas en multiples modules fournit un mécanisme au moyen duquel les composantes pouvant être réutilisées peuvent être importées le cas échéant sans devoir importer 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 empêche 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

18. 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 neutres de contexte (ou opérationnellement indépendantes) et partagées par au moins deux types de propriété industrielle.

Annexe I, page 8

19. Il y a trois différents niveaux de composante, à savoir les composantes de base, les composantes agrégées et les composantes document. Les composantes communes sont classées en composantes de base et agrégées et les composantes opérationnelles en composantes de base, en composantes agrégées et en composantes document.

Niveau	Description
Composante de base	<p>Une composante de base est utilisée par une ou plusieurs composantes agrégées. La composante de base est représentée par un élément.</p> <p>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>. Par exemple, <code>FirstName</code> (string), <code>PhoneNumberCategory</code> (<code>PhoneNumberCategoryType</code>).</p>
Composante agrégée	<p>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 ou non d'un contexte commercial spécifique. Elle est représentée par un élément et un type (types complexes qui comportent de multiples éléments).</p> <p>Une composante agrégée se réfère à la définition de la composante de base, à d'autres définitions de composantes agrégées ou à des normes externes mentionnées dans le présent document. Par exemple, <code>Name</code> (<code>NameType</code>), <code>Contact</code> (<code>ContactType</code>).</p>
Composante document	<p>Une composante document est une composante de haut niveau qui définit les principaux documents échangés. Elle est représentée par un élément et un type complexe.</p> <p>Une composante document se réfère à la définition de la composante de base, à la définition de la composante agrégée ou à une norme externe mentionnée dans le présent document. Par exemple, <code>ApplicationBody</code> (<code>ApplicationBodyType</code>), <code>MarkRecord</code> (<code>MarkRecordType</code>).</p>

20. Exemple de déclaration de la composante de base :

```
<?xml version="1.0" encoding="UTF-8"?>
<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:com="http://www.wipo.int/standards/XMLSchema/Common/0"
targetNamespace="http://www.wipo.int/standards/XMLSchema/Common/0"
elementFormDefault="qualified" attributeFormDefault="qualified" version="0.7">...
  <xsd:include schemaLocation="sequenceNumber-V0-5.xsd"/>
  <xsd:include schemaLocation="TextType-V0-5.xsd"/>
  <xsd:complexType name="AddressLineType">
    <xsd:annotation>
      <xsd:documentation>A line in an address. </xsd:documentation>
    </xsd:annotation>
    <xsd:simpleContent>
      <xsd:extension base="com:TextType">
        <xsd:attribute ref="com:sequenceNumber" use="required"/>
      </xsd:extension>
    </xsd:simpleContent>
  </xsd:complexType>
</xsd:schema>
```

2.3.2 Renvoi à des schémas externes

21. Chaque module d'un schéma réside dans un espace de nommage distinct. Dans le présent document, chaque module d'un schéma est défini dans un de quatre espaces de nommage qui sont réservés aux modules du schéma de la composante commune, de la composante brevet, de la composante marque et de la composante dessins et modèles industriels respectivement.

22. Deux structures de schéma W3C, c'est-à-dire, `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 à celles uniquement qui appartiennent à un espace de nommage particulier.

Annexe I, page 9

23. La figure 1 décrit la manière d'utiliser le modèle de modularité de la norme ST.96 de l'OMPI pour élaborer des schémas.

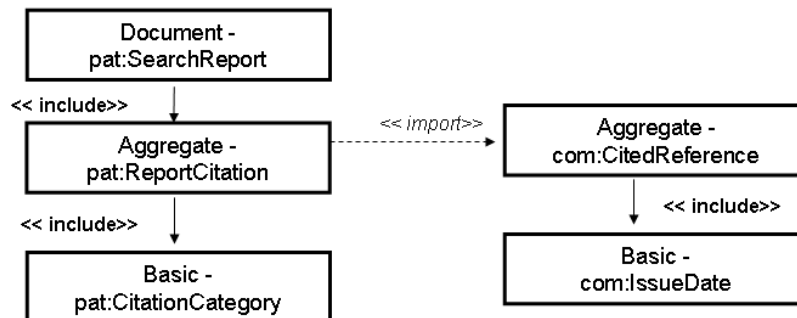


Figure 1. Exemple de modularité d'un schéma

- [SD-01] Les modules du schéma de la composante brevet, de la composante marque et de la composante dessins et modèles industriels DOIVENT utiliser la structure `import` pour désigner les modules du schéma de la composante commune.
- [SD-02] Le module du schéma de la composante commune NE DOIT PAS se référer aux modules du schéma de la composante brevet, de la composante marque et de la composante dessins et modèles industriels.
- [SD-03] Les modules du schéma de la composante brevet, de la composante marque et de la composante dessins et modèles industriels NE DOIVENT PAS se renvoyer l'un à l'autre.

2.4 Réutilisabilité

24. 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 le 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 qui prennent en compte le découplage et la cohésion dans les schémas XML.

25. 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.

26. 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'un document XML valide adhérant au schéma défini pour cet espace de nommage.

27. La plupart des schémas du monde réel font de Venetian Blind ou de Garden of Eden leur style de choix car ils sont le plus réutilisables. Aux fins du découplage et de la cohésion, Venetian Blind peut être meilleure que Garden of Eden mais, lorsqu'il s'agit de la réutilisabilité, Garden of Eden peut être meilleur que Venetian Blind en utilisant et le type et l'élément. 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 que soit adopté Garden of Eden comme style de conception.

- [SD-04] Les schémas 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 existants.
- [SD-06] Tous les types, éléments et attributs DOIVENT être déclarés globalement.

Annexe I, page 10

[SD-07] Les schémas NE DOIVENT PAS utiliser la structure `redefine` pour la nouvelle définition.

2.5 Espaces de nommage

28. Les espaces de nommage sont uniquement utilisés pour identifier 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. Un document 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 réserves 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 URL.

[SD-13] Tous les schémas DOIVENT avoir des attributs `elementFormDefault` et `attributeFormDefault` avec valeur "qualified"

29. Pour des raisons d'efficacité, les offices de propriété industrielle 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

30. Les noms des espaces de nommage sont importants pour la stratégie de contrôle des schémas. Il est essentiel que les règles de nommage des espaces de nommage soient suivies lors de la conception des schémas. Indépendamment du petit numéro de la version du schéma donné, l'espace de nommage doit uniquement contenir le grand numéro de la version. Par exemple : <http://www.wipo.int/standards/XMLSchema/Trademark/1>

31. On trouvera dans l'exemple suivant un espace de nommage bien formé :

```
<?xml version="1.0" encoding="UTF-8"?>
<xsd:schema xmlns:com="http://www.wipo.int/standards/XMLSchema/Common/1"
xmlns:xsd="http://www.w3.org/2001/XMLSchema">
.....
</xsd:schema>
```

32. De plus amples informations sont disponibles dans la section sur l'établissement des versions de schéma du présent document.

[SD-14] Les noms des espaces de nommage DOIVENT avoir la structure suivante : <http://www.wipo.int/standards/XMLSchema/<category>/<major>> où `<category>` est un symbole identifiant le domaine du schéma utilisé : Common, Patent, Trademark ou Design; et `<major>` est le grand nombre de la version.

[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 brevet, "tmk" pour tous les schémas de la composante marque, "dgn" pour tous les schémas de la composante dessins et modèles industriels, "tbl" pour les schémas de la composante OASIS Table et "mathml" pour les schémas de la composante MathML.

Annexe I, page 11

2.5.3 Espaces de nommage cibles

33. 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.

34. 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/Common/<major>>, 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/1"
targetNamespace="http://www.wipo.int/standards/XMLSchema/Common/1"
elementFormDefault="qualified" attributeFormDefault="qualified">
  <xsd:include schemalocation="xxx.xsd" />
</xsd:schema>
```

- <http://www.wipo.int/standards/XMLSchema/Patent/<major>>, 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/Patent/1"
targetNamespace="http://www.wipo.int/standards/XMLSchema/Patent/1"
elementFormDefault="qualified" attributeFormDefault="qualified">
  <xsd:include schemalocation="xxx.xsd" />
</xsd:schema>
```

- <http://www.wipo.int/standards/XMLSchema/Trademark/<major>>, 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/Trademark/1"
targetNamespace="http://www.wipo.int/standards/XMLSchema/Trademark/1"
elementFormDefault="qualified" attributeFormDefault="qualified">
  <xsd:include schemalocation="xxx.xsd" />
  ...
</xsd:schema>
```

- <http://www.wipo.int/standards/XMLSchema/Design/<major>>, 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/Design/1"
targetNamespace="http://www.wipo.int/standards/XMLSchema/Design/1"
elementFormDefault="qualified" attributeFormDefault="qualified">
  <xsd:include schemalocation="xxx.xsd" />
  .....
</xsd:schema>
```

[SD-16] Chaque schéma DOIT déclarer ses espaces de nommage cibles à l'aide de l'attribut `xsd:targetNamespace`.

[SD-17] Le `targetNamespace` du schéma doit correspondre au nom de l'espace de nommage d'une des qualifications déclarées mais pas la qualification du Consortium W3C.

Annexe I, page 12

- [SD-18] Les schémas de la composante commune DOIVENT utiliser <http://www.wipo.int/standards/XMLSchema/Common/<major>> comme l'espace de nommage cible.
- [SD-19] Les schémas de la composante brevet DOIVENT utiliser <http://www.wipo.int/standards/XMLSchema/Patent/<major>> comme l'espace de nommage cible.
- [SD-20] Les schémas de la composante marque DOIVENT utiliser <http://www.wipo.int/standards/XMLSchema/Trademark/<major>> comme l'espace de nommage cible.
- [SD-21] Les schémas de la composante dessins et modèles industriels DOIVENT utiliser <http://www.wipo.int/standards/XMLSchema/Design/<major>> comme l'espace de nommage cible.

2.5.4 Espaces de nommage implicites

35. 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. C'est pourquoi l'utilisation d'un espace de nommage implicite n'est pas recommandée dans la norme ST.96 de l'OMPI.

- [SD-22] Le schéma NE DEVRAIT PAS utiliser un espace de nommage implicite.

2.6 Versions de schéma

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

2.6.1 Modifications majeures et mineures

37. 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 documents créés avec des versions mineures antérieures de ce schéma avec la même version majeure. Toutefois, il ne faut pas s'attendre à ce que les documents valident des versions d'un schéma précédant celle avec laquelle ils ont été créés. En outre, le schéma peut faire l'objet de révisions lorsqu'il se trouve au stade de projet.

2.6.1.1 Versions majeures

38. La version majeure d'un schéma constitue une version ayant fait l'objet de modifications significatives et/ou non rétrocompatibles. Si un document XML fondé sur une version majeure plus ancienne d'un schéma cherche une validation par rapport à la nouvelle version, il risque de connaître des erreurs de validation. Une nouvelle version majeure sera établie lorsque des modifications significatives et/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

39. 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 versions mineures n'augmenteront que lorsque ont lieu des modifications compatibles comme par exemple :

Annexe I, page 13

- 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 tous les documents créés 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 projet du schéma n'est pas en mesure de valider les documents XML existants.
- [SD-25] Les versions mineures d'un schéma NE DEVRAIENT PAS être modifiées à moins que le schéma ou l'un des schémas importés/inclus ne soit altéré pour des modifications mineures.
- [SD-26] S'agissant des schémas, le nom de fichier du schéma, l'attribut `version` W3C et l'espace de nommage DOIVENT tous contenir des informations correspondant à la version.
- [SD-27] Les versions des schémas DEVRAIENT être modifiées lorsqu'un schéma inclus ou importé est actualisé.
- [SD-28] Lorsque la structure d'un schéma est altérée en raison des modifications majeures apportées dans un espace de nommage donné, tous les schémas dans cet espace DOIVENT faire l'objet d'une version majeure.
- [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 des schémas

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

- a) conventions de nommage des espaces des schémas;
- b) conventions de nommage des fichiers des schémas;
- c) utilisation de l'attribut `version` XML du schéma; et
- d) utilisation d'un attribut `schemaVersion` défini par l'utilisateur.

2.6.2.1 Espace de nommage dans le contrôle des schémas

41. D'après les règles de nommage des espaces de nommage qui figurent dans le présent document, les espaces de nommage ne contiennent qu'un certain nombre de versions majeures. Cela est important pour préserver la compatibilité entre les versions mineures du schéma. Étant donné que les fichiers plus anciens des versions mineures doivent continuer d'être validés par rapport aux versions mineures plus récentes du schéma, l'espace de nommage doit être conservé dans les versions mineures.

42. L'espace de nommage ne peut pas à lui seul servir à identifier facilement la version exacte d'un schéma ou d'un document donné. C'est ici que d'autres aspects du contrôle des schémas deviennent importants comme indiqué ci-dessous.

2.6.2.2 Conventions de nommage des fichiers dans le contrôle des schémas

43. Les informations sur les versions peuvent être incorporées dans le nom du fichier du schéma. D'après les règles de nommage des fichiers décrites dans le présent document, le nom du fichier du schéma contient un certain nombre de versions majeures et mineures.

Annexe I, page 14

2.6.2.3 Attribut incorporé de la version du schéma XML dans le contrôle des schémas

44. La spécification d'un schéma W3C permet la définition d'un attribut de `version` dans l'élément de base d'un schéma. Dans la norme ST.96 de l'OMPI, les schémas doivent inclure aussi bien un certain nombre de versions majeures qu'un certain nombre de versions mineures 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/Patent/1"
targetNamespace="http://www.wipo.int/standards/XMLSchema/Patent/1"
elementFormDefault="qualified" attributeFormDefault="unqualified" version="1.0">
...
</xsd:schema>
```

[SD-30] L'attribut `version` du schéma W3C DOIT inclure aussi bien un certain nombre de versions majeures qu'un certain nombre de versions mineures pour chaque fichier de schéma dans le format suivant : `<major version number>.<minor version number>`.

2.6.2.4 Attribut `schemaVersion` défini par l'utilisateur dans le contrôle des schémas pour les documents XML

45. L'attribut `schemaVersion` défini par l'utilisateur est nécessaire pour s'assurer qu'un document donné se réfère directement à une version spécifique du schéma. Cet attribut définira clairement la version du schéma que cible le document. Par exemple :

```
<?xml version="1.0"?>
<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema"
targetNamespace="http://www.wipo.int/standards/XMLSchema/Patent/1"
xmlns:pat="http://www.wipo.int/standards/XMLSchema/Patent/1"
elementFormDefault="qualified" attributeFormDefault="qualified" version="1.0">
<xsd:element name="SearchReport" type="SearchReportType"/>
<xsd:complexType name="SearchReportType"/>
  <xsd:attribute name="schemaVersion" type="xsd:String" use="required"/>
  .....
</xsd:complexType>
```

Le document XML correspondant :

```
<?xml version="1.0" encoding="UTF-8"?>
<pat:SearchReport xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:pat="http://www.wipo.int/standards/XMLSchema/Patent/1"
xsi:schemaLocation="http://www.wipo.int/standards/XMLSchema/Patent/1/Document
SearchReport-V1.0.xsd" schemaVersion="1.0">
.....
</pat:SearchReport>
```

[SD-31] Les schémas définis comme des composantes documents DOIVENT définir un attribut nécessaire appelé `schemaVersion` dans leur élément de base avec le format suivant : `<major version number>.<minor version number>`.

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

46. 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 industrielle. 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 industrielle et des fournisseurs d'information en matière de propriété industrielle, la norme ST.96 cherche à établir le degré nécessaire de possibilité de conversion avec les normes ST.36, ST.66 et ST.86 de l'OMPI.

Annexe I, page 15

47. Par possibilités de conversion, on entend que l'ensemble des unités d'information atomiques (niveau le plus bas des éléments de granularité, typiquement un élément sans aucun élément enfant) qui pourraient être incorporés dans un document XML conforme à la norme ST.96 peut être converti en unités d'information atomiques pouvant être incorporées dans un document XML conforme aux normes ST.36, ST.66 ou ST.86, et vice versa.

48. Les listes d'énumération devraient tenir compte des valeurs prévues pour les unités d'information atomiques respectives qui sont disponibles dans les normes ST.36, ST.66 ou ST.86 de l'OMPI. Elles devraient inclure les valeurs, si nécessaire, "Other", ce qui signifie que la valeur fournie dans les données d'origine n'est pas présente dans la liste énumérée, et "Undefined", ce qui signifie que l'information n'était pas présente dans les données d'origine (documents XML conformes aux normes ST.36, ST.66 et ST.86).

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

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

2.8 Schémas de normes industrielles

49. 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 industrielle, il faut utiliser des schémas de normes industrielles. Les schémas de normes industrielles qui sont recommandés dans le présent document sont le schéma de tableau OASIS XML et MathML. Le premier est disponible à l'adresse suivante : <http://www.oasis-open.org/docbook/xmlschema/1.0b1/calstbl.xsd> qui est un schéma modèle et qui doit être précisé davantage à des fins commerciales. Pour répondre aux besoins commerciaux de la propriété industrielle, dans le schéma ST.96, OASISTable.xsd est défini sur la base du schéma modèle. S'agissant des 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>.

50. Dans le cas des formules ou structures chimiques, il n'y avait ni une recommandation du Consortium W3C ni une norme industrielle XML largement convenue lorsque le présent document a été élaboré. 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 d'acceptation universelle. Par conséquent, le présent document ne recommande aucune norme XML pour les données chimiques.

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

[SD-35] Pour les tableaux, OASISTable.xsd qui repose sur le schéma OASIS XML DEVRAIT être utilisé.

[SD-36] Les schémas de normes industrielles DEVRAIENT être incorporés par voie de référence 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

51. 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 couplés.

3.1.1 Types simples

52. 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 lexicographique 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 W3C et les types de données définis par l'utilisateur. Dans la norme ST.96 de l'OMPI, les types simples sont définis comme les types de données définis par l'utilisateur.

Annexe I, page 16

3.1.1.1 Types de données incorporées du W3C

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

3.1.1.2 Types de données définis par l'utilisateur

54. Un des avantages des schémas XML est la capacité qu'ils offrent de définir les types de données des utilisateurs. Les types de données définis par l'utilisateur reposent sur les types de données incorporées du W3C et peuvent également être tirées des types de données existants définis par l'utilisateur. Ils peuvent être tirées 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 types simples DEVRAIENT être utilisés pour définir les listes de codes à l'aide d'une énumération.
- [SD-39] Les types simples PEUVENT utiliser `xsd:union` pour améliorer une liste de codes.
- [SD-40] La norme ST.3 de l'OMPI (codes à deux lettres) DOIT être utilisée pour la représentation des offices de propriété industrielle ainsi que pour les pays/organisations désignés et prioritaires. Par exemple, `PriorityCountryCode="EP"`.
- [SD-41] La norme ISO 3166-1-Alpha-2 Eléments 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 valeurs de date et d'heure dans les documents de schéma DOIVENT se conformer aux types de données relatives à la date et à l'heure du schéma XML 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 listes d'énumération DOIVENT se limiter aux caractères suivants : {a-z, A-Z, 0-9, point (.), virgule (,), espaces, tiret (-), esperluette (&#amp;#2262;) et caractère de soulignement (_)}.
- [SD-46] Les valeurs figurant dans les listes 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 industrielle.

3.1.2 Types complexes

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

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

3.2 Éléments et attributs

56. 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.

Annexe I, page 17

3.2.1 Éléments ou attributs

57. Une des principales décisions à prendre en matière de conception d'un schéma est celle qui porte sur la question de savoir 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 un attribut, il n'est pas possible de l'étoffer davantage. C'est pourquoi les attributs DEVRAIENT uniquement être utilisés 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 industrielle dans les documents XML. Les attributs DEVRAIENT détenir des métadonnées accessoires, c'est-à-dire des données simples qui fournissent plus d'informations sur l'élément.

[SD-48] Les schémas DEVRAIENT utiliser des attributs uniquement lorsqu'ils définissent des données non commerciales. Par exemple, l'attribut `sequenceNumber`.

3.2.2 Éléments

3.2.2.1 Cardinalité des éléments

58. Le terme cardinalité est défini comme étant le nombre d'éléments dans un ensemble.

59. 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] Les schémas NE DEVRAIENT PAS utiliser des indicateurs d'occurrence pour les valeurs implicites, c'est-à-dire `minOccurs="1"`; `maxOccurs="1"`.

3.2.2.2 Éléments vides

60. Un élément vide est un élément sans contenu, 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.

61. Bien que la norme ST.36 exploite les éléments vides comme étant un type d'indicateur `Boolean` qu'une condition était vraie ou fausse, selon la présence ou l'absence de l'élément, cette pratique est retirée dans la norme ST.96. Au contraire, des éléments devraient être créés qui affirment explicitement soit la condition vraie soit la fausse, éliminant tous les doutes ou 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

62. 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. C'est pourquoi les attributs ne peuvent pas être étendus comme les éléments.

63. La cardinalité pour les attributs est différente de celle pour les éléments. L'indicateur d'utilisation `use` peut être déterminé pour un attribut avec une des valeurs suivantes : "required", "optional" ou "prohibited".

[SD-51] Les schémas NE DEVRAIENT PAS employer pour un attribut l'indicateur d'utilisation `use` lorsque la valeur escomptée est la valeur implicite (c'est-à-dire `use="optional"`).

3.2.4 Regroupement des éléments et attributs

64. Les compositeurs sont les structures du schéma W3C qui regroupent les déclarations d'éléments. Il y a trois catégories de compositeurs dans la norme du schéma W3C, à savoir la `sequence`, `choice` et `all`.

65. Le compositeur `sequence` indique que les éléments qui y sont déclarés à l'intérieur doivent figurer dans un document XML dans l'ordre déclaré. Il permet l'application de l'ordre des éléments.

Annexe I, page 18

66. Le compositeur `choice` indique qu'un seul des éléments qui y sont déclarés à l'intérieur peut figurer dans un document XML.

67. 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 `sequence/choice` facultatif ou multiple, pour éviter la définition d'un niveau supplémentaire.

68. Le compositeur `all` indique que les éléments qui y sont déclarés à l'intérieur ne peuvent figurer qu'une seule fois dans un document 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*

69. 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. Ces techniques sont les suivantes : extension, restriction et groupes de substitution.

3.3.1 Extension

70. 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. Le fait de tirer des types simples par `union` ou par `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

71. Dans le contexte de l'harmonisation des données, la restriction de types complexes agit sur 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 agit sur les valeurs. Aucun problème n'est censé se poser lorsque sont échangées des données.

3.3.3 Groupes de substitution

72. Les groupes de substitution permettent à un élément global de remplacer un autre élément global dans un document XML sans devoir apporter des modifications additionnelles au schéma. Ils n'encouragent pas cependant l'harmonisation des noms d'élément. L'harmonisation est la clé de l'échange interopérable de données 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*

73. À l'image d'un système de stockage, un document XML doit offrir des moyens d'identifier des éléments d'information qu'il contient ou de s'y référer.

Il y a deux caractéristiques qui permettent au XML de le faire avec un schéma XML du Consortium W3C :

- `xsd:ID/xsd:IDREF/IDREFS` imite directement les types d'attribut `ID`, `IDREF` et `IDREFS` des DTD XML; et
- `xsd:key/xsd:keyref/xsd:unique` a été introduit pour fournir une plus grande souplesse au moyen d'expressions XPATH.

Annexe I, page 19

- [SD-56] Les schémas DEVRAIENT utiliser `xsd:key/xsd:keyref/xsd:unique` pour les contraintes en matière d'identité.
- [SD-57] Les schémas NE DEVRAIENT PAS utiliser `ID/IDREF/IDREFS` pour les contraintes en matière d'identité.

3.5 *Documentation du schéma*

74. La spécification du schéma W3C permet à la documentation d'être incluse dans les schémas. Cela permet aux schémas de contenir des descriptions incorporées de chaque structure. Une documentation est nécessaire pour communiquer le contexte et l'utilisation d'éléments dans les schémas à l'utilisateur qui peut ne pas avoir les mêmes antécédents ou compétences professionnelles que le développeur de schémas.

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

76. Il arrive que des commentaires du style HTML sont 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 de documentation W3C. La documentation DEVRAIT uniquement décrire l'élément ou le type et NE DEVRAIT PAS contenir des détails sur la mise en œuvre ou d'autres informations qui ne sont pas directement liées à la signification de la structure.
- [SD-59] Les commentaires (c'est-à-dire `<!-- comment -->`) NE DEVRAIENT PAS être utilisés dans un schéma.
- [SD-60] La documentation NE DEVRAIT PAS remplacer les listes de codes utilisant une énumération.

3.5.1 *Documentation relative à l'en-tête du schéma*

77. 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. Comme dans le cas de la documentation sur la structure d'un schéma, l'élément de documentation du schéma W3C devrait être utilisé pour la documentation relative à l'en-tête du schéma.

78. Le tableau 1 donne la liste des éléments qui doivent être inclus dans la section de l'en-tête de tous les schémas.

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

Nom de l'en-tête	Description
"Last update"	Date de la dernière mise à jour du schéma
"Contact Point"	Personne à contacter avec des questions sur le schéma
"Release Notes URL"	L'URL pour lequel les notes de diffusion du schéma sont publiées

Le nom de l'en-tête devrait précéder le contenu réel de l'en-tête comme dans l'exemple suivant :

```
<xsd:annotation>
  <xsd:documentation>
    Last update: 2011-05-28
    Contact point: xml.standards@wipo.int
    Release notes URL: http://www.wipo.int/standards/XMLSchema/ReleaseNotes.txt
  </xsd:documentation>
</xsd:annotation>
```

79. La documentation relative à l'en-tête du schéma permet à un développeur de schémas 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 DOIVENT inclure une documentation relative à leur en-tête qui contient des renvois à "Last update", "Contact point" et "Release notes URL".

Annexe I, page 20

4. RÈGLES DE CONCEPTION DES DOCUMENTS

80. Les règles de conception des documents traitent des recommandations à suivre lorsque sont fournies des données structurées.

4.1 *Espaces de nommage dans les documents XML*

4.1.1 Validation des documents XML

81. Il est recommandé de valider un document 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 XML.

82. 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.

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

```
<?xml version="1.0" encoding="UTF-8"?>
<com:Contact xmlns:com="http://www.wipo.int/standards/XMLSchema/Common/1"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://www.wipo.int/standards/XMLSchema/Common/1/Aggregat
e/Contact-V1-0.xsd">
.....
</com:Contact>
```

84. L'attribut `schemaLocation` est une structure de schéma W3C qui associe un document XML à un schéma. Il n'est utilisé que lorsqu'un schéma à un espace de nommage cible. Dans l'exemple ci-dessus, l'identificateur d'espace de nommage "`http://www.wipo.int/standards/XMLSchema/Common/1`" est l'espace de nommage cible du schéma `Contact.xsd`.

[ID-01] Les documents XML DOIVENT être validés en fonction du schéma correspondant.

[ID-02] Les documents XML DOIVENT utiliser l'attribut `schemaLocation` pour décrire l'espace de nommage et le nom du schéma auquel le document XML se conforme.

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

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

85. Comme dans le cas des schémas, un espace de nommage est déclaré dans l'élément de base d'un document XML à l'aide d'un identificateur d'espace de nommage avec un préfixe recommandé. Les documents 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.

86. Il sied de noter que :

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

87. Les éléments et attributs dans les documents 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 XML. C'est pourquoi tous les éléments et attributs doivent être qualifiés d'espaces de nommage. Toutefois, l'exigence pour les éléments et attributs 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`.

Annexe I, page 21

88. 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 chacun d'eux avoir la valeur `qualified` ou `unqualified` (par défaut).

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/Patent/1"
targetNamespace="http://www.wipo.int/standards/XMLSchema/Patent/1"
elementFormDefault="qualified"
attributeFormDefault="qualified">
.....
</xsd:schema>
```

89. 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 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, c'est-à-dire le préfixe "com" pour toutes les structures de schéma à composante commune, "pat" pour toutes les structures de schéma à composante brevet, "tmk" pour toutes les structures de schéma à composante marque, "dgn" pour toutes les structures de schéma à composante dessins et modèles industriels, "tbl" pour toutes les structures de schéma à composante tableau OASIS et "mathml" pour les structures de schéma à composante MathML.

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

4.1.3 L'espace de nommage du schéma W3C

90. 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 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 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/Common/1">
.....
</com:Contact>
```

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

[ID-06] Les documents 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

92. Les espaces de nommage dans les documents XML ont un champ d'applicabilité dans un document 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.

93. 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 en examinant l'élément de base est elle plus utile.

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

Annexe I, page 22

4.2 Entités externes

94. Les images incrustées constituent une entité externe utilisée couramment, c'est-à-dire qu'un renvoi à un fichier d'images externe est inséré dans un document à l'endroit où l'image devrait être montrée lorsque le document est établi. 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 notamment.

95. Un document XML peut renvoyer à un autre fichier XML fondé sur un DTD en tant qu'entité externe. Dans ce cas-là, il ne peut pas être validé contre le schéma XML correspondant. La possibilité de renvoi au fichier XML pourrait consister à utiliser `xsd:anyURI` ou `xsd:string` pour indiquer l'emplacement du fichier extérieur du document XML. Dans ce document cependant, il n'est pas recommandé de renvoyer au fichier XML fondé sur le DTD.

96. Les images peuvent être intégrées dans un document 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 à des fichiers d'images extérieurs, c'est-à-dire des entités externes. Dans le présent document cependant, les images doivent être désignées comme des entités externes car des d'images binaires incrustées peuvent inclure un code nuisible comme par exemple un virus.

- [ID-08] Les entités externes qui sont des images DEVRAIENT être conformes à un des formats d'image recommandés dans la norme ST.96 de l'OMPI.
- [ID-09] Les images DOIVENT être appelées fichiers extérieurs.
- [ID-10] Les listages de séquences DEVRAIENT suivre la norme ST.25 de l'OMPI.

Annexe I, page 23

APPENDICE A – RÉSUMÉ 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 (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]	La norme ISO/IEC 10646 – UCS – Unicode DOIT être utilisée pour l'ensemble des caractères. La norme UTF-8 DOIT être utilisée pour coder les caractères Unicode.
[GD-04]	Les noms de types, d'éléments et d'attributs DOIVENT être composés de mots anglais, écrits selon l'orthographe anglaise figurant dans le dictionnaire Oxford English Dictionary, y compris les noms propres à un office, à l'exception des sigles, des abréviations ou d'autres mots tronqués qui sont énumérés dans l'appendice C.
[GD-05]	Les noms de types, d'éléments et d'attributs DOIVENT uniquement contenir des noms, des adjectifs et des verbes au présent.
[GD-06]	Les caractères utilisés dans des noms de types d'éléments et d'attributs DOIVENT être limités à la série suivante : {a-z, A-Z et 0-9}.
[GD-07]	La longueur maximum d'un nom DEVRAIT être de 35 caractères.
[GD-08]	Les noms de types, d'éléments et d'attributs DEVRAIENT être concis et s'expliquer d'eux-mêmes.
[GD-09]	Les noms d'éléments DOIVENT figurer en caractères haut de casse de type "camel" (UCC). Par exemple, <code>CountryCode</code> .
[GD-10]	Les noms de types DOIVENT se présenter sous la forme UCC + type suffixe. Par exemple, <code>ApplicantType</code> .
[GD-11]	Les noms d'attributs DOIVENT figurer en caractères bas de casse de type "camel" (LCC). Par exemple, <code>currencyCode="EUR"</code> .
[GD-12]	Les sigles et abréviations indiqués dans 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 figurer en caractères de bas de casse. Tous les autres sigles et abréviations utilisés dans une déclaration d'attribut DOIVENT figurer en caractères de haut de casse.
[GD-14]	Les sigles et abréviations DOIVENT figurer en caractères de haut de casse pour tous les noms de type et d'élément.
[GD-15]	Les noms de type complexe DEVRAIENT inclure un terme de classe objet significatif.
[GD-16]	Les noms de type complexe d'association DEVRAIENT utiliser une structure de classe objet du type complexe associant (nature de l'association), et la classe objet du type complexe associé et des qualificatifs. Par exemple, <code>ApplicantResidenceAddress: Applicant</code> est le postulant d'une 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 de type simple et d'élément atomique (s'il n'a pas d'enfants) DEVRAIENT comporter le terme de classe objet, le terme de propriété, le terme de représentation et le terme de qualificatif.
[GD-18]	Un terme de classe objet DOIT toujours avoir la même signification sémantique dans un espace de nommage et PEUT avoir plus d'un mot. Par exemple, <code>ContactInformation</code> .
[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]	Le terme de classe objet DOIT occuper dans le nom la première place (la plus à gauche), le terme de propriété la place suivante et le terme de représentation la dernière place (la plus à droite). Le 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 alors être supprimé.

Annexe I, page 24

Règle ID	Règle
[GD-23]	Les termes de représentation à l'appendice B DOIVENT être utilisés pour les termes de représentation dans les noms de composante.
[GD-24]	Dans un espace de nommage, tous les noms de type, d'élément et d'attribut (qu'ils soient déclarés localement ou globalement) 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'une collection DEVRAIT utiliser le suffixe "Bag". Par exemple, <code>EmailAddressBag</code> représente une collection d' <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. Par exemple, <code>GoodsAndServices</code> .
[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 d'un schéma DOIVENT être limités à la série suivante : {a-z, A-Z, 0-9, tiret (-), et point (.)}.
[GD-31]	Le nom de fichier d'un schéma DOIT comporter cinq parties avec trois 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-V1-1.xsd</code> , <code>languageCode-V1-0.xsd</code> , <code>id-V1-2.xsd</code> .
[GD-32]	Le nom de fichier du projet de schéma DOIT comporter sept parties avec quatre délimiteurs, 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-V1-0-D3.xsd</code> .

Règles de conception d'un schéma

Règle ID	Règle
[SD-01]	Les modules du schéma de la composante brevet, de la composante marque et de la composante dessins et modèles industriels DOIVENT utiliser la structure <code>import</code> pour désigner les modules du schéma de la composante commune.
[SD-02]	Le module du schéma de la composante commune NE DOIT PAS se référer aux modules du schéma de la composante brevet, de la composante marque et de la composante dessins et modèles industriels.
[SD-03]	Les modules du schéma de la composante brevet, de la composante marque et de la composante dessins et modèles industriels NE DOIVENT PAS se renvoyer l'un à l'autre.
[SD-04]	Les schémas 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 existants.
[SD-06]	Tous les types, éléments et attributs DOIVENT être déclarés globalement.
[SD-07]	Les schémas NE DOIVENT PAS utiliser la structure <code>redefine</code> pour la nouvelle définition.
[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 réserves 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 URL.
[SD-13]	Tous les schémas DOIVENT avoir des attributs <code>elementFormDefault</code> et <code>attributeFormDefault</code> avec valeur "qualified".
[SD-14]	Les noms des espaces de nommage DOIVENT avoir la structure suivante : <code>http://www.wipo.int/standards/XMLSchema/<category>/<major></code> où <category> est un symbole identifiant le domaine du schéma utilisé : <code>Common</code> , <code>Patent</code> , <code>Trademark</code> ou <code>Design</code> ; et <major> est le numéro de la version majeure.

Annexe I, page 25

Règle ID	Règle
[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 brevet, "tmk" pour tous les schémas de la composante marque, "dgn" pour tous les schémas de la composante dessins et modèles industriels, "tbl" pour les schémas de la composante OASIS Table et "mathml" pour les schémas de la composante MathML.
[SD-16]	Chaque schéma DOIT déclarer ses espaces de nommage cibles 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'une des qualifications déclarées mais pas la qualification du Consortium W3C.
[SD-18]	Les schémas de la composante commune DOIVENT utiliser <code>http://www.wipo.int/standards/XMLSchema/Common/<major></code> comme l'espace de nommage cible.
[SD-19]	Les schémas de la composante brevet DOIVENT utiliser <code>http://www.wipo.int/standards/XMLSchema/Patent/<major></code> comme l'espace de nommage cible.
[SD-20]	Les schémas de la composante marque DOIVENT utiliser <code>http://www.wipo.int/standards/XMLSchema/Trademark/<major></code> comme l'espace de nommage cible.
[SD-21]	Les schémas de la composante dessins et modèles industriels DOIVENT utiliser <code>http://www.wipo.int/standards/XMLSchema/Design/<major></code> comme l'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 tous les documents créés 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 projet du schéma n'est pas en mesure de valider les documents XML existants.
[SD-25]	Les versions mineures d'un schéma NE DEVRAIENT PAS être modifiées à moins que le schéma ou l'un des schémas importés/inclus ne soit altéré pour des modifications mineures.
[SD-26]	S'agissant des schémas, le nom de fichier du schéma, l'attribut <code>version</code> W3C et l'espace de nommage DOIVENT tous contenir des informations correspondant à la version.
[SD-27]	Les versions des schémas DEVRAIENT être modifiées lorsqu'un schéma inclus ou importé est actualisé.
[SD-28]	Lorsque la structure d'un schéma est altérée en raison des modifications majeures apportées dans un espace de nommage donné, tous les schémas dans cet espace DOIVENT faire l'objet d'une version majeure.
[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 inclure aussi bien un certain nombre de versions majeures qu'un certain nombre de versions mineures pour chaque fichier de schéma dans le format suivant : <code><major version number>.<minor version number></code> .
[SD-31]	Les schémas définis comme des composantes documents DOIVENT définir un attribut nécessaire appelé <code>schemaVersion</code> dans leur élément de base avec le format suivant : <code><major version number>.<minor version number></code> .
[SD-32]	Les schémas ST.96 DEVRAIENT être élaborés avec un niveau similaire de cardinalité et de granularité afin de faciliter les possibilités de conversion des documents ST.96 avec les documents ST.36, ST.66 ou ST.86.
[SD-33]	Les énumérations contenues dans les schémas NE DEVRAIENT PAS inclure les valeurs "Other" ou "Undefined" à moins qu'elles ne soient nécessaires.
[SD-34]	Pour les formules mathématiques, MathML, version 3, DEVRAIT être utilisée.
[SD-35]	Pour les tableaux, OASISTable.xsd qui repose sur le schéma OASIS XML DEVRAIT être utilisé.
[SD-36]	Les schémas de normes industrielles DEVRAIENT être incorporés par voie de référence 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 types simples DEVRAIENT être utilisés pour définir les listes de codes à l'aide d'une énumération.
[SD-39]	Les types simples PEUVENT utiliser <code>xsd:union</code> pour améliorer une liste de codes.

Annexe I, page 26

Règle ID	Règle
[SD-40]	La norme ST.3 de l'OMPI (codes à deux lettres) DOIT être utilisée pour la représentation des offices de propriété industrielle 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 de langues.
[SD-43]	Les valeurs de date et d'heure dans les documents de schéma DOIVENT se conformer aux types de données relatives à la date et à l'heure du schéma XML 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 listes d'énumération DOIVENT se limiter aux caractères suivants : {a-z, A-Z, 0-9, point (.), virgule (,), espaces, tiret (-), esperluette (&#x26;) et caractère de soulignement (_)}.
[SD-46]	Les valeurs figurant dans les listes 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 industrielle.
[SD-47]	Des types complexes abstraits PEUVENT être utilisés.
[SD-48]	Les schémas DEVRAIENT utiliser des attributs uniquement lorsqu'ils définissent des données non commerciales. Par exemple, l'attribut <code>sequenceNumber</code> .
[SD-49]	Les schémas NE DEVRAIENT PAS utiliser des indicateurs d'occurrence pour les valeurs implicite (c'est-à-dire <code>minOccurs="1"</code> ; <code>maxOccurs="1"</code>).
[SD-50]	Les éléments vides NE DOIVENT PAS être définis dans les schémas, sauf pour le saut de ligne.
[SD-51]	Les schémas NE DEVRAIENT PAS employer pour un attribut l'indicateur <code>use</code> lorsque la valeur escomptée est la valeur implicite (c'est-à-dire, <code>use="optional"</code>).
[SD-52]	Les schémas NE DEVRAIENT PAS utiliser le compositeur <code>a11</code> .
[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> pour les contraintes en matière d'identité.
[SD-57]	Les schémas NE DEVRAIENT PAS utiliser <code>ID/IDREF/IDREFS</code> pour les contraintes en matière d'identité.
[SD-58]	Tous les schémas DEVRAIENT inclure une documentation sur leurs structures utilisant l'élément de documentation W3C. La documentation DEVRAIT uniquement décrire l'élément ou le type et NE DEVRAIT PAS contenir des détails sur la mise en œuvre 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><!--comment --></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 DOIVENT inclure une documentation relative à leur en-tête qui contient des renvois à la dernière mise à jour, au point de contact et aux notes de diffusion URL.
Règle ID	Règle
[ID-01]	Les documents XML DOIVENT être validés en fonction du schéma correspondant.
[ID-02]	Les documents XML DOIVENT utiliser l'attribut <code>schemaLocation</code> pour décrire l'espace de nommage et le nom du schéma auquel le document XML se conforme.
[ID-03]	Chaque déclaration d'attribut <code>xmlns:schemaLocation</code> DEVRAIT contenir un URL facile d'accès.

Annexe I, page 27

Règle ID	Règle
[ID-04]	Chaque espace de nommage déclaré DEVRAIT recevoir un préfixe recommandé, c'est-à-dire le préfixe "com" pour toutes les structures de schéma à composante commune, "pat" pour toutes les structures de schéma à composante brevet, "tmk" pour toutes les structures de schéma à composante marque, "dgn" pour toutes les structures de schéma à composante dessins et modèles industriels, "tbl" pour toutes les structures de schéma à composante tableau OASIS et "mathml" pour les structures de schéma à composante MathML.
[ID-05]	Les documents XML NE DOIVENT PAS utiliser des espaces de nommage implicites.
[ID-06]	Les documents XML DEVRAIENT utiliser xsi comme un préfixe d'espace de nommage pour toutes les structures de schéma W3C.
[ID-07]	Les documents XML NE DOIVENT PAS utiliser des déclarations locales d'espace de nommage.
[ID-08]	Les entités externes qui sont des images DEVRAIENT être conformes à un des formats d'image recommandés dans la norme ST.96 de l'OMPI.
[ID-09]	Les images DOIVENT être appelées fichiers extérieurs.
[ID-10]	Les listages de séquences DEVRAIENT suivre la norme ST.25 de l'OMPI.

Annexe I, page 28

APPENDICE B – TERMES DE REPRÉSENTATION

Terme	Définition	Data Type
Montant	Une valeur monétaire	AmountType
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.	xsd:token
Date	La notion d'un moment spécifique, désignée par l'année, le mois et le jour	DateType
Identificateur	Une combinaison d'un ou plusieurs nombres entiers, lettres, caractères spéciaux qui identifie le cas spécifique d'un objet commercial mais qui peut ne pas avoir une signification facile à définir.	xsd:token
Indicateur	Un signal de la présence, de l'absence ou de l'exigence de quelque chose. Les valeurs recommandées sont Y, N, et "?" en cas de besoin.	xsd:boolean
Mesure	Une mesure est une valeur numérique déterminée en mesurant un objet avec l'unité de mesure donnée. <i>MeasureType</i> 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, <i>MeasureType</i> devrait être utilisé pour mesurer les propriétés intrinsèques ou physiques d'un objet considéré comme un tout.	MeasureType
Nom	La désignation d'un objet exprimé en un mot ou en une phrase.	xsd:string
Nombre	Un chiffre ou une série de chiffres désignant le label, la valeur, la quantité ou l'identification.	xsd:positiveInteger OU xsd:string
Pour cent	Un chiffre qui représente la partie d'un tout qui sera divisé par 100.	xsd:decimal
Quantité	Une quantité est un nombre compté d'unités non monétaires, y compris éventuellement des fractions. <i>QuantityType</i> est utilisé pour représenter un nombre compté de choses. <i>QuantityType</i> 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. <i>QuantityType</i> devrait toujours exprimer un nombre compté de choses et la propriété sera totale, expédiée, chargée, stockée.	QuantityType
Taux	Une quantité ou un montant mesuré en fonction d'une autre quantité ou d'un autre montant.	xsd:decimal
Texte	Une chaîne de caractères non formatés, généralement sous la forme de mots (comprend les abréviations et commentaires).	xsd:string
Temps	La désignation d'un moment chronologique précis dans une période	xsd:time
Timestamp	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

Annexe I, page 29

APPENDICE C – LISTE DES SIGLES ET ABRÉVIATIONS

ALT	Texte alternatif pour une image
B	Caractère gras
BioDeposit	Dépôt biologique
Br	Saut de ligne
DD	Description de la définition
DL	Liste des définitions
DOI	Identificateur d'objet numérique
DT	Terme de définition
DTD	Définition de type de document
I	Italique
ID	Identificateur pour l'identification des systèmes
IDREF	Identifieur Reference
IP	Propriété industrielle
IPC	Classification internationale des brevets
IPO	Office de propriété industrielle
IPR	Droit de propriété industrielle
ISO	Organisation internationale de normalisation
LCC	Caractères bas de casse de type "camel"
LI	Liste
NPL	Littérature non-brevet
O	Over score
OCR	Reconnaissance optique des caractères
OL	Ordered List
P	Paragraphe
Pre	Texte préformaté
Sub	Indice
Sup	Exposant
SWIFT	Society for Worldwide Interbank Financial Telecommunication
U	Soulignement
UCC	Caractères haut de casse de type "camel"
UL	Unordered List
URI	Uniform Resource Identifier
URL	Uniform Resource Locator
URN	Uniform Resource Name
W3C	World Wide Web Consortium
WIPO	Organisation mondiale de la propriété intellectuelle
XML	eXtensible Markup Language

Annexe I, page 30

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
- Norme ST.25 de l'OMPI : Présentation du listage des séquences de nucléotides et d'acides aminés
- Norme ST.36 de l'OMPI : Utilisation du XML dans le traitement de l'information en matière de brevets
- 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.86 de l'OMPI : Utilisation du langage XML dans le traitement de l'information en matière de dessins et modèles industriels

Normes industrielles

- W3C XML Schema Definition Language (XSD), série de recommandations—Schéma XML, Première partie 1 : Structures et schéma XML; Deuxième partie 2 : Types
- Appel d'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 pays
- Norme ISO 639-1:2002 – Codes pour la représentation des noms de langue – Partie 1 : 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
- OASIS XML Table Schema : <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.

[Fin de l'annexe I]