

## **Comité de Normas Técnicas de la OMPI (CWS)**

### **Reanudación de la cuarta sesión Ginebra, 21 a 24 de marzo de 2016**

#### **PROPUESTA RELATIVA A LOS ANEXOS V Y VI DE LA NORMA ST.96 DE LA OMPI**

*Documento preparado por la Secretaría*

#### **INTRODUCCIÓN**

1. En su segunda sesión, celebrada en 2012, el Comité de Normas Técnicas de la OMPI (CWS) adoptó la Norma ST.96 de la OMPI, junto con los Anexos I a IV; tomó nota asimismo de que, con miras a su aprobación, era necesario seguir elaborando los proyectos de otros dos Anexos, el Anexo V (Directrices y normas de aplicación del esquema) y el Anexo VI (Directrices y normas sobre transformación), que se presentaron en la sesión. Por lo tanto, el CWS solicitó al Equipo Técnico XML4IP que preparase una propuesta para la aprobación por el CWS de esos Anexos como parte de la Norma ST.96 de la OMPI.
2. Desde 2011, el Equipo Técnico XML4IP ha mantenido numerosas rondas de debate con objeto de preparar la propuesta relativa a los Anexos V y VI. Gracias a la activa colaboración de todos los miembros del Equipo Técnico, la preparación de los Anexos V y VI ha finalizado.
3. Los proyectos finales de los Anexos V y VI que se someten al examen del CWS para su aprobación se basan en la versión 2.0 del esquema XML de la Norma ST.96. La propuesta relativa al Anexo V, que contiene dos Apéndices, figura en el Anexo I del presente documento y la propuesta relativa al Anexo VI, que contiene tres Apéndices, figura en el Anexo II del presente documento.

#### ANEXO V DE LA NORMA ST.96

4. El objetivo del Anexo V de la Norma ST.96, *Directrices y normas de aplicación del esquema*, es proporcionar pautas para adaptar los esquemas de XML definidos en el Anexo III de la Norma ST.96, de manera común, a fin de adaptarlos a las necesidades concretas de una oficina de P.I. (OPI), aunque, en muchos casos, el esquema XML de la Norma ST.96 puede ser utilizado sin necesidad de modificarlo. Sin embargo, es posible que las OPI tengan que hacer frente a situaciones de uso para las que los esquemas XML de la Norma ST.96 no cumplan con los requisitos exigidos, y el Anexo V pretende orientar a las OPI en la elaboración de sus propios esquemas basados en los de la Norma ST.96 y otros recursos XML.

5. Hay dos tipos de esquemas de adaptación, a saber, esquema compatible con la Norma ST.96 de la OMPI y esquema conforme a esa Norma. Una vez que una OPI decide adaptar un esquema de la Norma ST.96, deberá decidir si el resultado será *conforme* a la Norma ST.96 o *compatible* con la Norma ST.96. En el Anexo V se ofrece un amplio conjunto de normas y directrices para la adaptación y aplicación de los esquemas XML de la Norma ST.96. Además, para guiar a las OPI con ejemplos concretos, el Anexo V contiene dos Apéndices, a saber, el Apéndice A con ejemplos de esquemas conformes a la Norma ST.96 y el Apéndice B con ejemplos de esquemas compatibles con la Norma ST.96.

6. Asimismo, a partir de ahora, *ST96XSDValidator* forma parte del Anexo V. *ST96XSDValidator* es una herramienta que utiliza *Schematron* para validar un esquema XML con arreglo a las *Normas de diseño y convenciones de XML de la Norma ST.96* (Anexo I de la Norma ST.96). La herramienta resulta muy práctica para las OPI a la hora de validar su aplicación del esquema XML con arreglo a la Norma ST.96.

#### ANEXO VI DE LA NORMA ST.96

7. Antes de la adopción de la Norma ST.96 de la OMPI, las OPI ya utilizaban las Normas ST.36, ST.66 y ST.86 de la OMPI; por lo tanto, mantener la transformabilidad con las instancias XML conformes a esas Normas era una de las principales preocupaciones del Equipo Técnico XML4IP. Para facilitar el intercambio de datos y la interoperabilidad entre una OPI que utiliza las Normas ST.36/ST.66/ST.86 y una OPI que utiliza la Norma ST.96, era necesaria la capacidad de transformación entre la Norma ST.96 y las Normas existentes; fue por ese motivo que el Equipo Técnico XML4IP empezó a elaborar el Anexo VI.

8. El Anexo VI de la Norma ST.96 de la OMPI, *Directrices y normas sobre transformación*, tiene como finalidad proporcionar directrices y normas para permitir la transformación entre instancias XML conformes a la Norma ST.96 e instancias XML validadas con arreglo a las Normas ST.36, ST.66 o ST.86. Cabe observar que en el Anexo VI no se pretende abordar la transformación para su aplicación en cada país ni la transformación de las instancias XML entre versiones diferentes de la Norma ST.96. Con miras a orientar a las OPI sobre cómo convertir los datos de que disponen que sean conformes a las Normas ST.36, ST.66 y ST.86 en datos conformes a la Norma ST.96 y viceversa, en el Anexo VI figuran tres apéndices, a saber, el Apéndice A, con tablas de asignación de elementos y atributos; el Apéndice B, con tablas de asignación de elementos de enumeración y el Apéndice C, con códigos de ejemplo de plantillas de transformación de XSLT (*eXtensible Stylesheet Language Transformations*) basadas en las tablas de asignación de los Apéndices A y B. Los tres Apéndices están disponibles en el sitio web de la OMPI, en: [http://www.wipo.int/edocs/mdocs/cws/en/cws\\_4\\_bis/cws\\_4\\_bis\\_12-appendix1.zip](http://www.wipo.int/edocs/mdocs/cws/en/cws_4_bis/cws_4_bis_12-appendix1.zip).

## MODIFICACIÓN DE LA TAREA N° 41

9. Habida cuenta de que el Equipo Técnico XML4IP ha finalizado la preparación de la propuesta relativa a los Anexos V y VI de la Norma ST.96, la Oficina Internacional propone suprimir la primera parte de la Tarea N° 41 y mantener solo su segunda parte. El nuevo texto de la Tarea N° 41 sería el siguiente “Tarea N° 41: garantizar las necesarias revisiones y actualizaciones de la Norma ST.96 de la OMPI”. La Oficina Internacional propone asimismo que el Equipo Técnico XML4IP prosiga el trabajo en relación con la Tarea N° 41 revisada.

10. *Se invita al CWS a:*

a) *tomar nota de la información contenida en el presente documento;*

b) *examinar y aprobar el Anexo V de la Norma ST.96 de la OMPI, que se reproduce en el Anexo I del presente documento;*

c) *examinar y aprobar el Anexo VI de la Norma ST.96 de la OMPI, que se reproduce en el Anexo II del presente documento y los tres Apéndices del Anexo VI de la Norma ST.96 propuesto, según se menciona en el párrafo 8; y*

d) *aprobar la modificación de la Tarea N° 41 y la asignación de la Tarea N° 41 revisada al Equipo Técnico XML4IP, según se propone en el párrafo 9 del presente documento.*

[Sigue el Anexo I]

## ST.96 - ANNEX V

### IMPLEMENTATION RULES AND GUIDELINES

Final Draft

*Proposal presented by the XML4IP Task Force for consideration and adoption at the CWS/4BIS*

#### Table of Contents

1. INTRODUCTION .....	2
1.1 Overview.....	2
1.2 Scope .....	2
1.3 How to use this document.....	2
1.4 Terminology .....	2
1.5 Rule identifiers .....	2
2. DESIGNING FOR ST.96 CUSTOMIZATION .....	3
2.1 Designing for conformance .....	3
2.2 Designing for compatibility .....	4
3. IMPLEMENTATION.....	6
3.1 Schema folder.....	6
3.2 Customized schemas.....	6
3.3 Using the ST.96 components for non-ST.96 document types .....	6
3.4. Testing implementation for conformation with DRCs .....	7
4. DESIGN-STAGE AND FLATTENED SCHEMAS.....	7
5. REFERENCES .....	8
APPENDIX A.....	9
1. Subset of the schema model .....	9
APPENDIX B.....	21
1. Complex types schemas.....	21

## 1. INTRODUCTION

### 1.1 Overview

1. WIPO Standard ST.96 provides a collection of XML resources that, for many cases, can be used “as is”. However, Intellectual Property Offices (IPOs) may need to address use cases whose requirements are not met by the XML Schemas defined in ST.96. This document is intended to guide how IPOs develop their schemas based on ST.96 Schemas and other XML resources.

2. One of the goals of the ST.96 Implementation Guidelines is to promote the implementation of ST.96 Schemas in a common way by IPOs and to maintain a consistent usage of element and attribute names between specific implementations.

### 1.2 Scope

3. The scope of this document is to provide a comprehensive set of rules and guidelines for the customization and implementation of ST.96 Schemas.

### 1.3 How to use this document

4. This document is intended for use by IPOs which want to implement and customize ST.96 Schemas. This document refers to the main body of ST.96 and its Annexes I, II and III as prerequisites.

### 1.4 Terminology

5. In this document:

- the term “Customization” refers to the alteration of ST.96 Schemas in order to better fit specific requirements.
- the term “ST.96 compatible schema” refers to a schema consistent with the ST.96 Schema Components and the XML Design Rules and Conventions for Industrial Property (DRCs), i.e. Annex I of ST.96. XML instances valid for the ST.96 compatible schema are not guaranteed to successfully validate against ST.96 Schemas.
- the term “ST.96 conformant schema” refers to a compatible schema that has not been extended and that sustains constraints expressed by an ST.96 Schema. XML instances valid for an ST.96 conformant schema will validate against ST.96 Schemas.
- the term “Basic Component” refers to W3C Built-in Datatype or simple Type or complex Type with `xsd:simpleContent` definition.
- the term “Aggregate Component” refers to the collection of related Basic Components and/or other Aggregate Components that together convey a distinct business meaning, independent or not of any specific business context.

### 1.5 Rule identifiers

6. All Office Implementation rules are informative and are identified through a prefix of [OI nn]. The value “nn” indicates the sequential number of the rule. For example, the rule identifier [OI-06] identifies the 6th rule of the Office Implementation rules.

### 1.6 Overview of ST.96 Schema Development Procedure

7. The ST.96 Component Schemas have been expressed in W3C XML Schema (XSD), based on the ST.96 DRCs. The ST.96 Component Schemas are used to both specify and validate ST.96 conformance. The following steps are presented as the procedure for developing ST.96 Schema components and a possible procedure for developing Office implementation schema.

- Step 1: Analyze the business processes; identify information to be exchanged, used and stored in the processes and define data for the information;
- Step 2: Model the data defined in Step 1;
- Step 3: Look for previously developed XML schema components that can be reused;
- Step 4: Create the ST.96 compliant name and definition for each component following XML Design Rules and Conventions (DRCs, ST.96 Annex I);
- Step 5: Identify any common business terms that are associated with the components;

- Step 6: Build schema components; and
- Step 7: Validate schema component using Schematron.

## 2. DESIGNING FOR ST.96 CUSTOMIZATION

8. The design of the conceptual models for ST.96 and its customizations are not affected by the syntactical issues of XML, schema languages, or validation tools. Once the need to customize the ST.96 Schema has been determined, IPOs SHOULD decide whether the result will be ST.96 *conformant* or ST.96 *compatible* schema. It should be noted that the ST.96 Schemas and conformant schemas are acceptable for data exchange.

9. Schematron can be used to validate whether an IPO's implementation schema follows the XML Design Rules and Conventions of ST.96. A sample Schematron, which contains some of the ST.96 design rules, is available in WIPO website at: <http://www.wipo.int/standards/en/st96/v2-0/annex-v/>

10. Designing a customization may involve:

- adding components to meet the requirements of a specific business context;
- omitting optional components not needed in a specific context;
- modifying constraints on possible values for basic components such as code lists; and
- combining (or recombining) and assembling components into new aggregate components or document components.

### 2.1 *Designing for conformance*

11. ST.96 conformance at the instance and schema level means that there are no constraint violations when validating the instance against the ST.96 Schema. When designing an XML schema for ST.96 conformance, the key objective is to create custom models that can be used to specify and validate ST.96 conformant instances. An ST.96 conformant instance is an instance that validates against an ST.96 Schema. An ST.96 conformant schema is a schema that validates only ST.96 conformant instances. An ST.96 conformant schema is a subset of an ST.96 Schema. Consequently, designing for conformance applies primarily to restrictions:

- subsets of the Schema model — restricting the number of elements in a Schema; and
- constraints on content — restricting the possible values a component can have.

12. There are no changes to the element references which import or include the conformant schemas.

#### 2.1.1 Subsets of the Schema model

13. The use of subsets allows for the removal from a component of any optional components that are not needed to satisfy the specific business requirements of an implementation.

14. It must be noted that sub-setting can only be used to remove optional elements or change cardinality in ways that do not reduce the required minimum number of occurrences or extend the permitted maximum number of occurrences of an element. The resulting range must be a subset of the original range. Thus, where  $1 < m < n$ ,

- 0..1 can become 1..1 or the element can be removed (but not, for example, 1..2);
- 0..n can become 0..1, 1..m, 1..n, m..n, or the element can be removed;
- 1..n can become 1..1, m..n, or 1..m; and
- 1..1 cannot be changed.

#### 2.1.2 Enumeration list constraints on content

15. Constraining the values for a component to a fixed set is a common customization requirement. For example, "the Currency Code must be expressed using ISO 4217 codes" is a constraint on the possible values for Currency Code in an instance.

- [OI-01] Office Implementation Schemas must not use `xsd:list` or `xsd:union` to derive simple Types defined in WIPO Standard ST.96.

#### 2.1.3 Other constraints on content

16. There are other cases in which the treatment of ST.96 instances may require customization in order to limit or restrict content values. For example, "The length of an Address Line cannot exceed 40 characters."

17. There are many business rules a customization may require that constrain the values used in the documents including co-occurrence. For example, the value of one or more components is affected by the value of one or more other components in the document content. Some of these constraints cannot be specified using schema validation semantics. Using Schematron, a customization can specify such assertions in a declarative fashion independent of how the assertions are actually implemented.

## 2.2 *Designing for compatibility*

18. While ST.96 conformance cannot be guaranteed, some degree of familiarity can be expected through the re-use of components defined in ST.96 (Annex III) and ST.96 Design Rules and Conventions (DRCs, Annex I). When creating new components or extending existing ST.96 Components, the DRCs MUST be followed to ensure compatibility.

### 2.2.1 Re-use of ST.96 Schema Components

19. Re-using ST.96 Schema Components keeps customization as closely aligned with ST.96 as possible and prevents an unnecessary proliferation of Components requiring maintenance. Do not recreate aggregate Components that already exist in ST.96.

### 2.2.2 Compatible extension of ST.96

20. If re-use of existing ST.96 Components is not feasible, customization can be made to add additional Components to the ST.96 Schema to satisfy business requirements. In these situations it is possible to extend ST.96 Components in a compatible manner.

21. *Extension* means to add to, or associate with, existing Components additional information that may be required for a particular context, for example, an extension creates a superset of the original Component. It is recommended that such an extension includes the original Component as an association from the Component that extends it. For example, `CitedIPDocumentType` defined in ST.96 is an extension of `CitedDocumentType` in ST.96 because it contains additional information required if the cited document is an IP document. Structurally, `CitedIPDocumentType` has an association to `CitedDocumentType`, making `CitedIPDocumentType` a superset of `CitedDocumentType`.

22. Compatible extensions can be implemented in parts of a schema. This allows validation checks to be built into the compatible schema that cannot be enforced in the extension area of a conformant schema.

23. The compatible schema does not share the same namespace as ST.96 Schemas. It allows the Office Implementation Schemas to use the default namespace. They should import ST.96 Schemas. To avoid confusion with ST.96 Schemas, it is not recommended to redefine them. If redefined, the namespace prefix for an Office's Implementation Schema should follow the two-letter codes (in lower case only) in WIPO Standard ST.3.

- [OI-02] Compatible Schemas should use the import construct to refer to ST.96 Schemas.
- [OI-03] In Compatible Schemas, redefinition of any types, elements and attributes defined in the WIPO Standard ST.96 using `xsd:redefine` construct should be avoided.
- [OI-04] Compatible Schemas may use a default namespace for efficient operations.
- [OI-05] Compatible Schemas must use the two-letter codes defined in WIPO Standard ST.3 for the office namespace prefix.

#### 2.2.2.1 Using qualified names

24. Besides namespace qualification, IPOs may also qualify the property term of a component specified as its Dictionary Entry Name in the IP Data Dictionary, i.e. Annex II of ST.96, to indicate that the customized component is based on an ST.96 Component. For example, "Address. Geographic Region" in ST.96 can be qualified for the Canadian office as "Address. CA Geographic Region", indicating that the use of the geographic region value is restricted to the context of Canadian provinces. The element name would be `CAGeographicRegion` which uses `GeographicRegionType` defined in ST.96. For example, `ruImageContentCategory` is an office implementation attribute name which is based on the ST.96 attribute `ImageContentCategory`.

- [OI-06] Compatible Schemas may use the two-letter codes defined in WIPO Standard ST.3 as a qualifier for customized components based on the ST.96 Schema.

#### 2.2.2.2 Re-using Aggregate Components

25. The principle applied is that if a required aggregate component has the same structure as a standard ST.96 component, then it should not be a redefinition but a re-use by association. The qualifying terms used to name the new associated component then describe the role it plays. For example, if an `AddressBook` is required for an Applicant, and this uses the normal `AddressBookType` structure, it could be defined as `ApplicantAddressBook`.

26. If the new aggregate component does not have the same structure as a standard ST.96 Component, then the required component MUST have a new name, not a qualified name. If possible, the new aggregate component may then be associated with the ST.96 Component being extended. For example, if an Address has additional components when the address is in Korea, then a new aggregate component called *KRAddress* would be created. This is not a qualification, but a new name. Ideally this should contain the original Address structure by association plus the new Korean components.

#### 2.2.2.3 New Basic components

27. A customization may require new basic components; these should be based on an existing ST.96 or W3C built-in data type (or a refinement thereof). Note that, where the new basic component is included in an aggregate component, it will result in a new aggregate component being defined as well.

28. When establishing a new basic component, it is necessary to associate it with a data type. This is determined by the Representation Term part of component's Dictionary Entry Name. For example, a Korean Address may have an additional component called "Dong. Text". This new basic information entity would use the standard Text data type.

29. Changing or specializing a component's definition changes the component. Therefore, a new basic component must be defined. For example, in ST.96, "Person. Other Name. Name" is defined as "Person name other than first name, middle name, last name, first last name, second last name, suffix and prefix". If a component is required to specify the Login name as a specific other name, then a new component (perhaps called "Person. Login Name. Name") should be defined.

30. In cases where the required component's representation does not fit an existing data type, a new data type may be required. New data types can be based on either ST.96 data types or W3C built-in data types. The following operations are allowed on ST.96 simple Types and examples of XML schema codes are available in Appendix A:

- add a character length restriction;
- add/change an enumeration list; and
- add a pattern restriction.

#### 2.2.2.4 New associations

31. Aggregate components are included in a schema model by associating them with a parent aggregate. If the required aggregation has the same structure as an existing aggregate a new association should be created with the existing aggregate (as in 2.2.2.2). This new association represents a new use of the aggregate, so qualifying terms can be used to describe the new role. For example, in ST.96, *AddressBook* is re-used in contexts such as *ClaimantAddressBook*, *ApplicantAddressBook* and *LicenseeAddressBook*. They all share the same structure as *AddressBookType* with the terms "Claimant", "Applicant", and "Licensee" providing the qualification.

32. By re-using the unqualified aggregate (*AddressBook*), the same XML type (*AddressBookType*) will be used for implementation of all these components.

#### 2.2.2.5 New Aggregate Components

33. A new aggregate component should be created if the required aggregation component does not exist in ST.96 or is an extension of an existing aggregate component, making it no longer conformant. When creating new aggregate components, there are some general principles to follow:

(a) A new aggregate component may also include the aggregate component being extended, as a child by extension. For example, in ST.96, *CitedIPDocumentType* is a new aggregate that has a different structure to *CitedDocumentType*. The *CitedDocumentType* structure is re-used by extension in *CitedIPDocumentType*, in addition, *CitedIPDocumentType* also contains additional components. The name *CitedIPDocumentType* is not a qualification of the name *CitedDocumentType*, but an extension to the ST.96 *CitedDocumentType* to create a new aggregate.

(b) New aggregate components should attempt to re-use patterns of ST.96 structures where possible.

(c) When defining new aggregate components, different constructs and mechanisms are not allowed in ST.96 Schemas but are tolerated in Office Implementation Schemas. This is the case for the all compositor and substitution groups.

(d) The following operations can be performed to customize an existing aggregate component. Examples of XML schema codes are provided in Appendix A:

- add and/or remove an element;
- add and/or remove an attribute;
- make an element mandatory and/or change its cardinality; and
- make an attribute required.



[OI-07] Compatible Schemas may use substitution groups.

#### 2.2.2.6 New document component schemas

34. Where existing document component schemas defined in ST.96 do not meet requirements, it is necessary to create a new document model. The key steps in assembling new document type structures are:

- (a) Select/create the root aggregate for the document type;
- (b) Assemble the required ST.96 Components (and/or customized extensions), applying cardinality constraints;
- (c) For all required associations from these Components assemble the required components (and/or customized extensions), applying cardinality constraints; and
- (d) Continue step (c) above recursively through all required associations.

### 3. IMPLEMENTATION

#### 3.1 Schema folder

35. It is recommended that the folder structure for implementation schemas be defined as it is in ST.96 to minimize changes of "import" declaration.

#### 3.2 Customized schemas

36. At least two scenarios in particular lend themselves to XSD derivations performed on existing types:
- an existing ST.96 Type fits the requirements for the application with modifications supported by XSD derivation. These modifications can include extension (adding new information to an existing type) and/or refinement (restricting the set of information allowed to a subset of that permitted by the existing type); and
  - no existing ST.96 type is found that can be used as the basis for the new type. Nevertheless, the base library of components that underlies ST.96 can be used to build up the new type so as to ensure that interoperability is at least possible at the basic component level.

37. Where the requirements are for a pure subset, it is possible to prune a ST.96 schema to create a new, smaller schema defining only the subset required. ST.96 relies on a common library of re-usable types, and therefore this approach does not support the restriction of selective types based on context. For example, an Address when used in one part of the subset schema cannot have a different restriction to an Address in another part of the document.

38. One approach for producing subset schemas is to work with the ST.96 schemas as input and use the XML comment construct to comment out all of the Components not used by the customization. A human reader of the schema specifications can see all of the ST.96 standardized constructs, easily distinguishing those that are in the customization and those that are not.

39. Another approach for producing subset schemas is to work at an abstract model level and to synthesize the schema fragments from scratch from the subset model. This approach implies all required elements are defined as global. In the following example, the first approach is applied.

```
<xsd:complexType name="PersonNameType">
  <xsd:sequence>
    <xsd:element ref="com:PersonNamePrefix" minOccurs="0"/>
    <xsd:element ref="com:FirstName" />
    <xsd:element ref="com:MiddleName" minOccurs="0"/>
    <xsd:element ref="com:LastName" />
    <xsd:element ref="com:FirstLastName" minOccurs="0"/>
    <xsd:element ref="com:SecondLastName" minOccurs="0"/>
    <xsd:element ref="com:PersonNameSuffix" minOccurs="0"/>
    <xsd:element ref="com:PreferredName" minOccurs="0"/>
    <xsd:element ref="com:PersonOtherName" minOccurs="0"/>
  </xsd:sequence>
</xsd:complexType>
```

#### 3.3 Using the ST.96 components for non-ST.96 document types

40. Even when a completely new document type must be defined, it can prove advantageous to use as many of the ST.96 Components as possible. Figure 1 shows an approach to specifying the schema fragments defining a non-ST.96 document using both ST.96 and non-ST.96 Components.

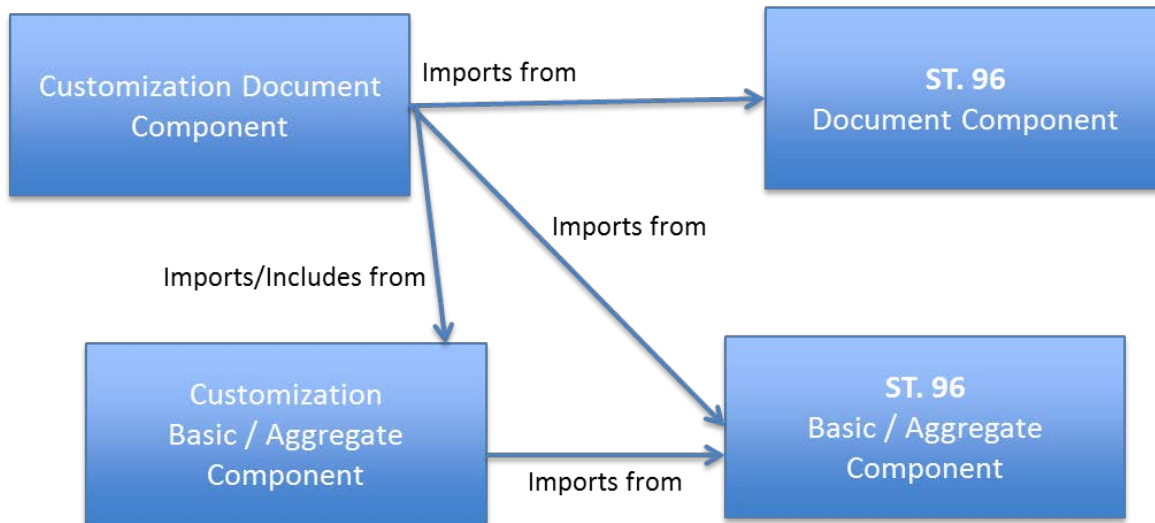


Figure 1. Using the ST.96 Components for non-ST.96 documents

#### 3.4. Testing implementation for conformance with DRCs

41. Rule-based validation such as Schematron may be used to validate an Office's Implementation Schema for conformance with DRCs (Annex I of ST.96).

#### 4. DESIGN-STAGE AND PRODUCTION-STAGE SCHEMAS

42. Depending on an IPO's requirements, production-stage schema can be different from the design-stage schema. The design-stage schemas may exist in two forms:

- Modular Schemas: Each element, simple type or complex type definition i.e. basic components and aggregate components reside in their own schema file. The advantage of using modular schemas is the ability to compose the larger schema (by means of Includes or Imports) and the smaller files are more manageable and reusable than the single full schema.
- Flattened Schemas: Flattened schemas contain all the dependent files that are included in a schema belonging to the same namespace into one file. In the process, the components of any included schemas are added as global components of the flattened schema. In this way, the flattened schema can be used without having to rely on a network connection or an IO process to access the required files. This could aid in reducing processing time for each instance.

43. The design-stage schema, in general, is used to produce the production-stage schema for efficiency purposes, which can be the flattened schemas in most cases. It must be ensured that the XML instances for data exchange conform to the design-stage schema, which means that the instances should contain the same namespace prefixes and schema version.

44. In a design-stage schema, all namespaces should be identified and elements and attributes fully qualified; all components should have their own version information. In the course of designing a schema based on the ST.96 Standard it is important to keep careful track of components using namespace and version information. To produce production-stage schemas it might be advisable to resolve all external references in a schema in advance and use the fully integrated schema in a production environment to reduce processing time for each instance. Such a practice is completely dependent on local circumstances and therefore no further recommendations can be given in this document.

45. Figure below shows how ST.96 Design Stage and ST.96 Production Schemas are leveraged to create Office Implementation Schemas for both Conformance and Compatibility. ST.96 Design Rules apply to both ST.96 Component Schemas and Office Implementation Conformant and Compatible Schemas. ST.96 design-stage and production-stage schemas can be imported to develop Office Implementation Schemas. The developed schemas can be validated for adherence to ST.96 Design Rules using Schematron.

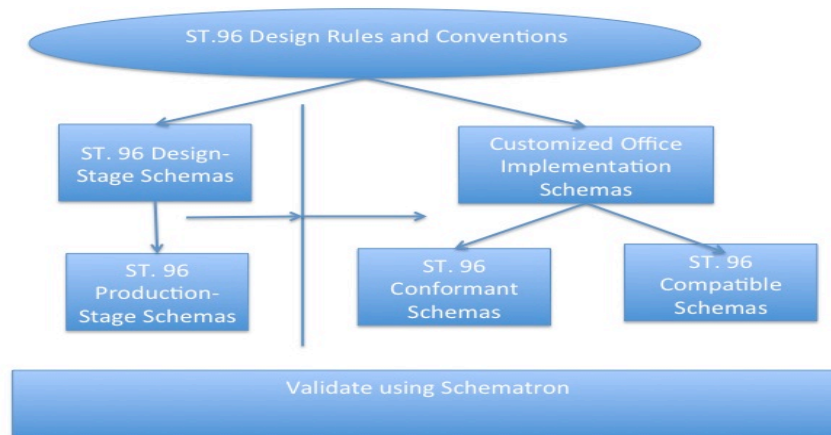


Figure2. Overview of ST.96 customization process

## 5. REFERENCES

- UBL 2 Guidelines for Customization, available at: <http://docs.oasis-open.org/ubl/guidelines/UBL2-Customization1.0cs01.pdf>
- Exchange Network shared schema components- usage guide, available at: [http://www.exchangenetwork.net/dev\\_schema/SharedSchemaComponents-UsageGuide\\_v2.0.pdf](http://www.exchangenetwork.net/dev_schema/SharedSchemaComponents-UsageGuide_v2.0.pdf)
- Guidelines for Electronic Text Encoding and Interchange- the TEI Consortium, P5, Using the TEI section, available at: <http://www.tei-c.org/release/doc/tei-p5-doc/en/html/USE.html>
- *Rule-based validation – Schematron*, ISO/IEC 19757-3:2006

## APPENDIX A - EXAMPLES OF ST.96 CONFORMANT SCHEMAS

The following examples are listed to show how developers define ST.96 conformant schemas. Designing for conformance applies primarily to restrictions which are:

- Subsets of the Schema model — restricting the number of Components in a Schema; and
- Constraints on content — restricting the possible values a Component can have.

### 1. SUBSET OF THE SCHEMA MODEL

#### 1.1 *Design-Stage Schemas - Modular Schemas: Remove an optional element*

#### **Patent Sample Business Case: Remove optional element UPOVCode from the PlantName Structure.**

1. Create a folder called "USPatent" at the same level as "Patent", "Trademark", "Common" etc.
2. Under USPatent, create subfolders "Document"
3. Under "USPatent" folder, create a new W3C schema file for Type, PlantNameType.xsd.
4. Set the target Namespace to the IPO's namespace and add new namespace entries `xmlns:xxpat`, where `xx` is the lower case ST.3 Code. Each IPO is expected to version the modified components separate from WIPO because this is the Office specific implementation. "V1\_0" is shown in the following example because this is the first Office modified component.

Please note that the examples in the Appendix use URN for IPO namespace as approved at USPTO. Other IPO's can use either URN or URL namespace based on their internal practice.

```
<?xml version="1.0" encoding="UTF-8"?>
<xsd:schema xmlns:com="http://www.wipo.int/standards/XMLSchema/ST96/Common"
xmlns:pat="http://www.wipo.int/standards/XMLSchema/ST96/Patent"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:uspat="urn:us:gov:doc:uspto:patent"
targetNamespace="urn:us:gov:doc:uspto:patent" elementFormDefault="qualified"
attributeFormDefault="qualified" version="V1_0">
```

5. Import ST.96 Components which are needed to define a new Type `PlantNameType`.

```
<xsd:import
namespace="http://www.wipo.int/standards/XMLSchema/ST96/Common"
schemaLocation="../Common/id.xsd"/>
<xsd:import
namespace="http://www.wipo.int/standards/XMLSchema/ST96/Patent"
schemaLocation="../Patent/BotanicalName.xsd"/>
<xsd:import
namespace="http://www.wipo.int/standards/XMLSchema/ST96/Patent"
schemaLocation="../Patent/PlantCommonName.xsd"/>
```

6. Declare the new Complex Type `PlantNameType` as shown below, which refers to existing elements with the exception of `UPOVCode`.

```
<xsd:complexType name="PlantNameType">
  <xsd:choice maxOccurs="unbounded">
    <xsd:element ref="pat:BotanicalName"/>
    <xsd:element ref="pat:PlantCommonName"/>
  </xsd:choice>
  <xsd:attribute ref="com:id"/>
</xsd:complexType>
</xsd:schema>
```

7. Update ST.96 schema file, `PlantName.xsd` in the Patent folder.

8. Declare IPO namespace. Each IPO is expected to version the modified components separate from WIPO because this is the Office specific implementation. "V1\_0" is shown in the following example because this is the first Office modified component.

```
<?xml version="1.0" encoding="UTF-8"?>
<xsd:schema xmlns:com="http://www.wipo.int/standards/XMLSchema/ST96/Common"
xmlns:pat="http://www.wipo.int/standards/XMLSchema/ST96/Patent"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:uspat="urn:us:gov:doc:uspto:patent"
targetNamespace="http://www.wipo.int/standards/XMLSchema/ST96/Patent"
elementFormDefault="qualified" attributeFormDefault="qualified"
version="V1_0">
```

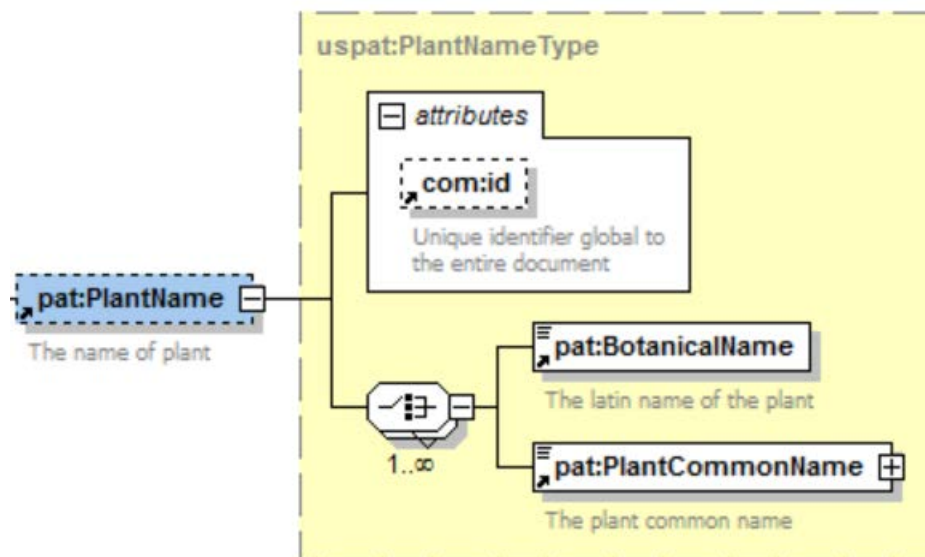
9. Import the schema file in which the new IPO Type was defined, PlantNameType.xsd.

```
<xsd:import namespace="urn:us:gov:doc:uspto:patent"
schemaLocation="..\USPatent\PlantNameType.xsd"/>
```

10. Update new IPO namespace reference to corresponding Type above.

```
<xsd:element name="PlantName" type="uspat:PlantNameType">
  <xsd:annotation>
    <xsd:documentation>The name of plant</xsd:documentation>
  </xsd:annotation>
</xsd:element>
</xsd:schema>
```

11. The final content model design is shown below:



1.2 *Design-Stage Schemas - Modular Schemas: Make an element required*

**Patent Sample Business Case: Make ClaimTotalQuantity required in the BibliographicData Structure.**

1. Create a folder called "USPatent" at the same level as "Patent", "Trademark", "Common" etc.
2. Create a new W3C schema file for Type called BibliographicDataType.xsd in folder "USPatent" --> "Document".
3. Set the target Namespace to the IPO's namespace and add the new namespace xmlns:xpat, where xx is an ST.3 Code.

```
<?xml version="1.0" encoding="UTF-8"?>
<xsd:schema xmlns:com="http://www.wipo.int/standards/XMLSchema/ST96/Common"
xmlns:pat="http://www.wipo.int/standards/XMLSchema/ST96/Patent"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:uspat="urn:us:gov:doc:uspto:patent"
targetNamespace="urn:us:gov:doc:uspto:patent" elementFormDefault="qualified"
attributeFormDefault="qualified" version="V1_0">
```

4. Import the ST.96 Components which are needed to define a new Type BibliographicDataType.

```
<xsd:import
namespace="http://www.wipo.int/standards/XMLSchema/ST96/Patent"
schemaLocation="../../Patent/PatentPublicationIdentification.xsd"/>
<xsd:import
namespace="http://www.wipo.int/standards/XMLSchema/ST96/Patent"
schemaLocation="../../Patent/ApplicationIdentification.xsd"/>
<xsd:import
namespace="http://www.wipo.int/standards/XMLSchema/ST96/Patent"
schemaLocation="../../Patent/PatentGrantIdentification.xsd"/>
<xsd:import
namespace="http://www.wipo.int/standards/XMLSchema/ST96/Patent"
schemaLocation="../../Patent/PriorityClaimBag.xsd"/>
<xsd:import
namespace="http://www.wipo.int/standards/XMLSchema/ST96/Patent"
schemaLocation="../../Patent/GrantTerm.xsd"/>
<xsd:import
namespace="http://www.wipo.int/standards/XMLSchema/ST96/Patent"
schemaLocation="../../Patent/PreClassificationText.xsd"/>
<xsd:import
namespace="http://www.wipo.int/standards/XMLSchema/ST96/Patent"
schemaLocation="../../Patent/PatentClassificationBag.xsd"/>
<xsd:import
namespace="http://www.wipo.int/standards/XMLSchema/ST96/Patent"
schemaLocation="../../Patent/InventionTitleBag.xsd"/>
<xsd:import
namespace="http://www.wipo.int/standards/XMLSchema/ST96/Patent"
schemaLocation="../../Patent/PlantName.xsd"/>
<xsd:import
namespace="http://www.wipo.int/standards/XMLSchema/ST96/Patent"
schemaLocation="../../Patent/ReferenceCitationBag.xsd"/>
<xsd:import
namespace="http://www.wipo.int/standards/XMLSchema/ST96/Patent"
schemaLocation="../../Patent/ClaimTotalQuantity.xsd"/>
<xsd:import
namespace="http://www.wipo.int/standards/XMLSchema/ST96/Patent"
schemaLocation="../../Patent/SearchField.xsd"/>
<xsd:import
namespace="http://www.wipo.int/standards/XMLSchema/ST96/Patent"
schemaLocation="../../Patent/FigureBag.xsd"/>
<xsd:import
namespace="http://www.wipo.int/standards/XMLSchema/ST96/Patent"
schemaLocation="../../Patent/RelatedDocumentBag.xsd"/>
<xsd:import
namespace="http://www.wipo.int/standards/XMLSchema/ST96/Patent"
schemaLocation="../../Patent/ProvisionalGrantPublicationDate.xsd"/>
<xsd:import
namespace="http://www.wipo.int/standards/XMLSchema/ST96/Patent"
schemaLocation="../../Patent/GrantPublicationDate.xsd"/>
<xsd:import
namespace="http://www.wipo.int/standards/XMLSchema/ST96/Patent"
schemaLocation="../../Patent/ExemplaryClaimBag.xsd"/>
```

```
<xsd:import
namespace="http://www.wipo.int/standards/XMLSchema/ST96/Patent "
schemaLocation="../../Patent/PartyBag.xsd" />
<xsd:import
namespace="http://www.wipo.int/standards/XMLSchema/ST96/Patent "
schemaLocation="../../Patent/InternationalFilingData.xsd" />
<xsd:import
namespace="http://www.wipo.int/standards/XMLSchema/ST96/Patent "
schemaLocation="../../Patent/RegionalFilingData.xsd" />
<xsd:import
namespace="http://www.wipo.int/standards/XMLSchema/ST96/Patent "
schemaLocation="../../Patent/InternationalPublishingData.xsd" />
<xsd:import
namespace="http://www.wipo.int/standards/XMLSchema/ST96/Patent "
schemaLocation="../../Patent/RegionalPublishingData.xsd" />
<xsd:import
namespace="http://www.wipo.int/standards/XMLSchema/ST96/Patent "
schemaLocation="../../Patent/PatentFamily.xsd" />
<xsd:import
namespace="http://www.wipo.int/standards/XMLSchema/ST96/Common "
schemaLocation="../../Common/BioDeposit.xsd" />
<xsd:import
namespace="http://www.wipo.int/standards/XMLSchema/ST96/Patent "
schemaLocation="../../Patent/PlainLanguageDesignationText.xsd" />
<xsd:import
namespace="http://www.wipo.int/standards/XMLSchema/ST96/Common "
schemaLocation="../../Common/FilingLanguageCode.xsd" />
<xsd:import
namespace="http://www.wipo.int/standards/XMLSchema/ST96/Common "
schemaLocation="../../Common/PublicationLanguageCode.xsd" />
<xsd:import
namespace="http://www.wipo.int/standards/XMLSchema/ST96/Common "
schemaLocation="../../Common/HagueAgreementData.xsd" />
<xsd:import
namespace="http://www.wipo.int/standards/XMLSchema/ST96/Patent "
schemaLocation="../../Patent/StateDesignation.xsd" />
<xsd:import
namespace="http://www.wipo.int/standards/XMLSchema/ST96/Patent "
schemaLocation="../../Patent/LicenceDataBag.xsd" />
<xsd:import
namespace="http://www.wipo.int/standards/XMLSchema/ST96/Patent "
schemaLocation="../../Patent/EarliestPriorityApplication.xsd" />
<xsd:import
namespace="http://www.wipo.int/standards/XMLSchema/ST96/Patent "
schemaLocation="../../Patent/CompleteSpecificationFilingDate.xsd" />
<xsd:import
namespace="http://www.wipo.int/standards/XMLSchema/ST96/Patent "
schemaLocation="../../Patent/PublicAvailabilityDateBag.xsd" />
<xsd:import
namespace="http://www.wipo.int/standards/XMLSchema/ST96/Patent "
schemaLocation="../../Patent/RightsEffectiveDateBag.xsd" />
<xsd:import
namespace="http://www.wipo.int/standards/XMLSchema/ST96/Patent "
schemaLocation="../../Patent/PatentDocumentRepublication.xsd" />
<xsd:import
namespace="http://www.wipo.int/standards/XMLSchema/ST96/Patent "
schemaLocation="../../Patent/ExhibitionFilingDate.xsd" />
<xsd:import
namespace="http://www.wipo.int/standards/XMLSchema/ST96/Patent "
schemaLocation="../../Patent/RightsReestablishedDate.xsd" />
<xsd:import
namespace="http://www.wipo.int/standards/XMLSchema/ST96/Patent "
schemaLocation="../../Patent/ApplicationWithdrawnDate.xsd" />
<xsd:import
namespace="http://www.wipo.int/standards/XMLSchema/ST96/Patent "
schemaLocation="../../Patent/ApplicationDeemedWithdrawnDate.xsd" />
<xsd:import
namespace="http://www.wipo.int/standards/XMLSchema/ST96/Patent "
schemaLocation="../../Patent/PatentRevocationDate.xsd" />
<xsd:import
namespace="http://www.wipo.int/standards/XMLSchema/ST96/Patent "
schemaLocation="../../Patent/PCTNationalPhaseEntryDate.xsd" />
```

```

<xsd:import
namespace="http://www.wipo.int/standards/XMLSchema/ST96/Patent"
schemaLocation="../../Patent/ApplicationPartiallyWithdrawnDate.xsd"/>
<xsd:import
namespace="http://www.wipo.int/standards/XMLSchema/ST96/Patent"
schemaLocation="../../Patent/OppositionData.xsd"/>
<xsd:import
namespace="http://www.wipo.int/standards/XMLSchema/ST96/Common"
schemaLocation="../../Common/id.xsd"/>
<xsd:import
namespace="http://www.wipo.int/standards/XMLSchema/ST96/Common"
schemaLocation="../../Common/officeCode.xsd"/>
<xsd:import
namespace="http://www.wipo.int/standards/XMLSchema/ST96/Common"
schemaLocation="../../Common/st96Version.xsd"/>
<xsd:import
namespace="http://www.wipo.int/standards/XMLSchema/ST96/Common"
schemaLocation="../../Common/ipoVersion.xsd"/>

```

5. Declare the new Complex Type BibliographicDataType which refers to existing elements and remove the minOccurs="0" attribute for the element ClaimTotalQuantity.

```

<xsd:complexType name="BibliographicDataType">
  <xsd:sequence>
    <xsd:element ref="pat:PatentPublicationIdentification"
minOccurs="0"/>
    <xsd:element ref="pat:ApplicationIdentification"/>
    <xsd:element ref="pat:PatentGrantIdentification"
minOccurs="0"/>
    <xsd:element ref="pat:PriorityClaimBag" minOccurs="0"/>
    <xsd:element ref="pat:GrantTerm" minOccurs="0"/>
    <xsd:element ref="pat:PreClassificationText" minOccurs="0"/>
    <xsd:element ref="pat:PatentClassificationBag"
minOccurs="0"/>
    <xsd:element ref="pat:InventionTitleBag"/>
    <xsd:element ref="pat:PlantName" minOccurs="0"/>
    <xsd:element ref="pat:ReferenceCitationBag" minOccurs="0"/>
    <xsd:element ref="pat:ClaimTotalQuantity"/>
    <xsd:element ref="pat:SearchField" minOccurs="0"/>
    <xsd:element ref="pat:FigureBag" minOccurs="0"/>
    <xsd:element ref="pat:RelatedDocumentBag" minOccurs="0"/>
    <xsd:element ref="pat:ProvisionalGrantPublicationDate"
minOccurs="0"/>
    <xsd:element ref="pat:GrantPublicationDate" minOccurs="0"/>
    <xsd:element ref="pat:ExemplaryClaimBag" minOccurs="0"/>
    <xsd:element ref="com:HagueAgreementData" minOccurs="0"/>
    <xsd:element ref="pat:PartyBag"/>
    <xsd:element ref="pat:InternationalFilingData"
minOccurs="0"/>
    <xsd:element ref="pat:RegionalFilingData" minOccurs="0"/>
    <xsd:element ref="pat:InternationalPublishingData"
minOccurs="0"/>
    <xsd:element ref="pat:RegionalPublishingData" minOccurs="0"/>
    <xsd:element ref="pat:PatentFamily" minOccurs="0"/>
    <xsd:element ref="com:BioDeposit" minOccurs="0"/>
    <xsd:element ref="pat:PlainLanguageDesignationText"
minOccurs="0"/>
    <xsd:element ref="com:FilingLanguageCode" minOccurs="0"/>
    <xsd:element ref="com:PublicationLanguageCode"
minOccurs="0"/>
    <xsd:element ref="pat:StateDesignation" minOccurs="0"/>
    <xsd:element ref="pat:LicenceDataBag" minOccurs="0"/>
    <xsd:element ref="pat:EarliestPriorityApplication"
minOccurs="0"/>
    <xsd:element ref="pat:CompleteSpecificationFilingDate"
minOccurs="0"/>
    <xsd:element ref="pat:PublicAvailabilityDateBag"
minOccurs="0"/>
    <xsd:element ref="pat:RightsEffectiveDateBag" minOccurs="0"/>
    <xsd:element ref="pat:PatentDocumentRepublication"
minOccurs="0"/>
    <xsd:element ref="pat:ExhibitionFilingDate" minOccurs="0"/>
  
```



```

        <xsd:element ref="pat:RightsReestablishedDate"
minOccurs="0"/>
        <xsd:element ref="pat:ApplicationWithdrawnDate"
minOccurs="0"/>
        <xsd:element ref="pat:ApplicationDeemedWithdrawnDate"
minOccurs="0"/>
        <xsd:element ref="pat:PatentRevocationDate" minOccurs="0"/>
        <xsd:element ref="pat:PCTNationalPhaseEntryDate"
minOccurs="0"/>
        <xsd:element ref="pat:ApplicationPartiallyWithdrawnDate"
minOccurs="0"/>
        <xsd:element ref="pat:OppositionData" minOccurs="0"/>
    </xsd:sequence>
    <xsd:attribute ref="com:id"/>
    <xsd:attribute ref="com:officeCode"/>
    <xsd:attribute ref="com:st96Version" use="required"/>
    <xsd:attribute ref="com:ipoVersion"/>
</xsd:complexType>
</xsd:schema>

```

6. Update ST.96 schema file, BibliographicData.xsd, in the Patent->Document folder.

7. Declare new IPO namespace.

```

<xsd:schema xmlns:com="http://www.wipo.int/standards/XMLSchema/ST96/Common"
xmlns:pat="http://www.wipo.int/standards/XMLSchema/ST96/Patent"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:uspat="urn:us:gov:doc:uspto:patent"
targetNamespace="http://www.wipo.int/standards/XMLSchema/ST96/Patent"
elementFormDefault="qualified" attributeFormDefault="qualified"
version="V1_0">

```

8. Update ST.96 BibliographicDataType.xsd include to import statement with the new IPO declared Type, BibliographicDataType.xsd.

```

<xsd:import namespace="urn:us:gov:doc:uspto:patent"
schemaLocation="../../../USPatent/Document/BibliographicDataType.xsd"/>

```

9. Update BibliographicDataType reference with new IPO namespace.

```

<xsd:element name="BibliographicData" type="uspat:BibliographicDataType">
    <xsd:annotation>
        <xsd:documentation>Bibliographic information included on the
first page of a patent
document. Contains document identification, domestic
filing data, foreign priority
data, public availability dates or term of protection,
technical information,
related patent or application
information</xsd:documentation>
    </xsd:annotation>
    <xsd:unique name="BibComponentKey">
        <xsd:selector xpath="//*[@*]" />
        <xsd:field xpath="@com:id" />
    </xsd:unique>
</xsd:element>

```

10. The final content model design is shown below:



1.3 *Design-Stage Schemas - Flattened Schemas: Remove an optional element*

**Patent Sample Business Case: Remove optional element UPOVCode from the PlantName Structure.**

1. Create a new W3C schema file, USBibliographicData\_V1\_0.xsd in the same folder as BibliographicData\_V2\_0.xsd.
2. Set the target Namespace to the IPO's namespace and add new namespace entries xmlns:xpat, where xx is the lower case ST.3 Code. Each IPO is expected to version the modified components separate from WIPO because this is the Office specific implementation. "V1\_0" is shown in the following example because this is the first Office modified component.

```
<?xml version="1.0" encoding="UTF-8"?>
<xsd:schema xmlns:com="http://www.wipo.int/standards/XMLSchema/ST96/Common"
xmlns:pat="http://www.wipo.int/standards/XMLSchema/ST96/Patent"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:uspat="urn:us:gov:doc:uspto:patent"
targetNamespace="urn:us:gov:doc:uspto:patent" elementFormDefault="qualified"
attributeFormDefault="qualified" version="V1_0">
```

3. Import ST.96 Components which are needed to define a new Type PlantNameType.

```
<xsd:import namespace="http://www.wipo.int/standards/XMLSchema/ST96/Common"
schemaLocation="Common_V2_0.xsd"/>
<xsd:import
namespace="http://www.wipo.int/standards/XMLSchema/ST96/Patent"
schemaLocation="BibliographicData_V2_0.xsd"/>
```

4. Create the new Complex Type PlantNameType as shown below, which refers to existing elements with the exception of UPOVCode.

```
<xsd:complexType name="PlantNameType">
<xsd:choice maxOccurs="unbounded">
<xsd:element ref="pat:BotanicalName"/>
<xsd:element ref="pat:PlantCommonName"/>
</xsd:choice>
<xsd:attribute ref="com:id"/>
</xsd:complexType>
</xsd:schema>
```

5. In the existing BibliographicData\_V2\_0.xsd file, declare the new schema file created above USBibliographicData\_V1\_0.xsd namespace. Each IPO is expected to version the modified components separate from WIPO because this is the Office specific implementation. "V1\_0" is shown in the below example because this is the first Office modified component.

```
<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:pat="http://www.wipo.int/standards/XMLSchema/ST96/Patent"
xmlns:com="http://www.wipo.int/standards/XMLSchema/ST96/Common"
xmlns:uspat="urn:us:gov:doc:uspto:patent"
targetNamespace="http://www.wipo.int/standards/XMLSchema/ST96/Patent"
elementFormDefault="qualified" attributeFormDefault="qualified"
version="V1_0">
```

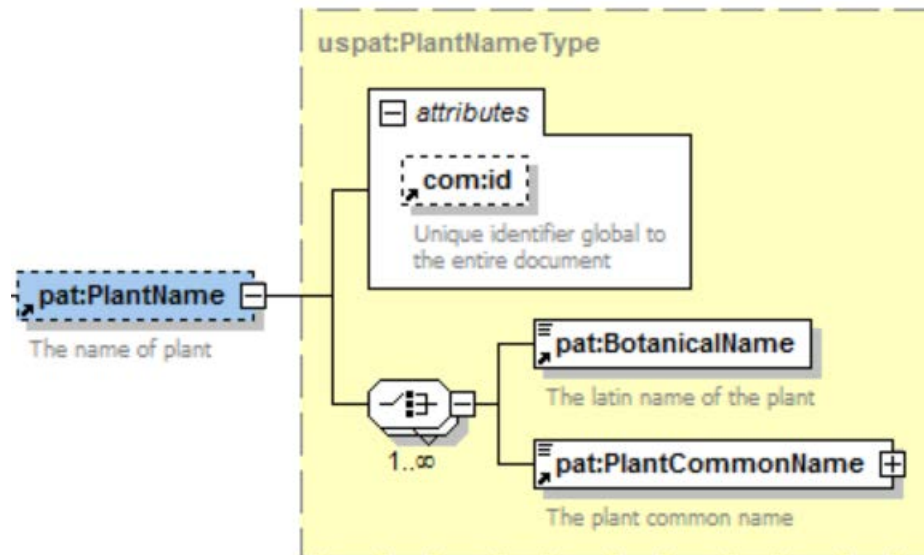
6. Import the new schema file USBibliographicData\_V1\_0.xsd in which the new IPO Type was defined, USBibliographicData\_V1\_0.xsd.

```
<xsd:import namespace="urn:us:gov:doc:uspto:patent"
schemaLocation="USBibliographicData_V1_0.xsd"/>
```

7. Update the element `PlantName` based on the corresponding new Type above with new namespace reference.

```
<xsd:element name="PlantName" type="uspat:PlantNameType">
  <xsd:annotation>
    <xsd:documentation>The name of plant</xsd:documentation>
  </xsd:annotation>
</xsd:element>
```

8. The final content model design for `PlantName` is shown below:



#### 1.4 Design-Stage Schemas - Flattened Schemas: Make an element required

##### Patent Sample Business Case: Make `ClaimTotalQuantity` required in the `BibliographicData` Structure.

1. Create a new W3C schema file, `USBibliographicData_V1_0.xsd` in the same folder as `BibliographicData_V2_0.xsd`.
2. Set the target Namespace to the IPO's namespace and add new namespace entries `xmlns:xspat`, where `xx` is the lower case ST.3 Code. Each IPO is expected to version the modified components separate from WIPO because this is the Office specific implementation. "V1\_0" is shown in the following example because this is the first Office modified component.

```
<?xml version="1.0" encoding="UTF-8"?>
<xsd:schema xmlns:com="http://www.wipo.int/standards/XMLSchema/ST96/Common"
xmlns:pat="http://www.wipo.int/standards/XMLSchema/ST96/Patent"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:uspat="urn:us:gov:doc:uspto:patent"
targetNamespace="urn:us:gov:doc:uspto:patent" elementFormDefault="qualified"
attributeFormDefault="qualified" version="V1_0">
```

3. Import ST.96 Components which are needed to define a new Type `BibliographicDataType`.

```
<xsd:import namespace="http://www.wipo.int/standards/XMLSchema/ST96/Common"
schemaLocation="Common_V2_0.xsd"/>
<xsd:import
namespace="http://www.wipo.int/standards/XMLSchema/ST96/Patent"
schemaLocation="BibliographicData_V2_0.xsd"/>
```

4. Create the new Complex Type BibliographicDataType which refers to existing elements and remove the minOccurs="0" attribute for the element ClaimTotalQuantity.

```

<xsd:complexType name="BibliographicDataType">
  <xsd:sequence>
    <xsd:element ref="pat:PatentPublicationIdentification"
minOccurs="0"/>
    <xsd:element ref="pat:ApplicationIdentification"/>
    <xsd:element ref="pat:PatentGrantIdentification"
minOccurs="0"/>
    <xsd:element ref="pat:PriorityClaimBag" minOccurs="0"/>
    <xsd:element ref="pat:GrantTerm" minOccurs="0"/>
    <xsd:element ref="pat:PreClassificationText" minOccurs="0"/>
    <xsd:element ref="pat:PatentClassificationBag"
minOccurs="0"/>
    <xsd:element ref="pat:InventionTitleBag"/>
    <xsd:element ref="pat:PlantName" minOccurs="0"/>
    <xsd:element ref="pat:ReferenceCitationBag" minOccurs="0"/>
    <xsd:element ref="pat:ClaimTotalQuantity"/>
    <xsd:element ref="pat:SearchField" minOccurs="0"/>
    <xsd:element ref="pat:FigureBag" minOccurs="0"/>
    <xsd:element ref="pat:RelatedDocumentBag" minOccurs="0"/>
    <xsd:element ref="pat:ProvisionalGrantPublicationDate"
minOccurs="0"/>
    <xsd:element ref="pat:GrantPublicationDate" minOccurs="0"/>
    <xsd:element ref="pat:ExemplaryClaimBag" minOccurs="0"/>
    <xsd:element ref="com:HagueAgreementData" minOccurs="0"/>
    <xsd:element ref="pat:PartyBag"/>
    <xsd:element ref="pat:InternationalFilingData"
minOccurs="0"/>
    <xsd:element ref="pat:RegionalFilingData" minOccurs="0"/>
    <xsd:element ref="pat:InternationalPublishingData"
minOccurs="0"/>
    <xsd:element ref="pat:RegionalPublishingData" minOccurs="0"/>
    <xsd:element ref="pat:PatentFamily" minOccurs="0"/>
    <xsd:element ref="com:BioDeposit" minOccurs="0"/>
    <xsd:element ref="pat:PlainLanguageDesignationText"
minOccurs="0"/>
    <xsd:element ref="com:FilingLanguageCode" minOccurs="0"/>
    <xsd:element ref="com:PublicationLanguageCode"
minOccurs="0"/>
    <xsd:element ref="pat:StateDesignation" minOccurs="0"/>
    <xsd:element ref="pat:LicenceDataBag" minOccurs="0"/>
    <xsd:element ref="pat:EarliestPriorityApplication"
minOccurs="0"/>
    <xsd:element ref="pat:CompleteSpecificationFilingDate"
minOccurs="0"/>
    <xsd:element ref="pat:PublicAvailabilityDateBag"
minOccurs="0"/>
    <xsd:element ref="pat:RightsEffectiveDateBag" minOccurs="0"/>
    <xsd:element ref="pat:PatentDocumentRepublication"
minOccurs="0"/>
    <xsd:element ref="pat:ExhibitionFilingDate" minOccurs="0"/>
    <xsd:element ref="pat:RightsReestablishedDate"
minOccurs="0"/>
    <xsd:element ref="pat:ApplicationWithdrawnDate"
minOccurs="0"/>
    <xsd:element ref="pat:ApplicationDeemedWithdrawnDate"
minOccurs="0"/>
    <xsd:element ref="pat:PatentRevocationDate" minOccurs="0"/>
    <xsd:element ref="pat:PCTNationalPhaseEntryDate"
minOccurs="0"/>
    <xsd:element ref="pat:ApplicationPartiallyWithdrawnDate"
minOccurs="0"/>
    <xsd:element ref="pat:OppositionData" minOccurs="0"/>
  </xsd:sequence>
  <xsd:attribute ref="com:id"/>
  <xsd:attribute ref="com:officeCode"/>
  <xsd:attribute ref="com:st96Version" use="required"/>
  <xsd:attribute ref="com:ipoVersion"/>
</xsd:complexType>
</xsd:schema>

```

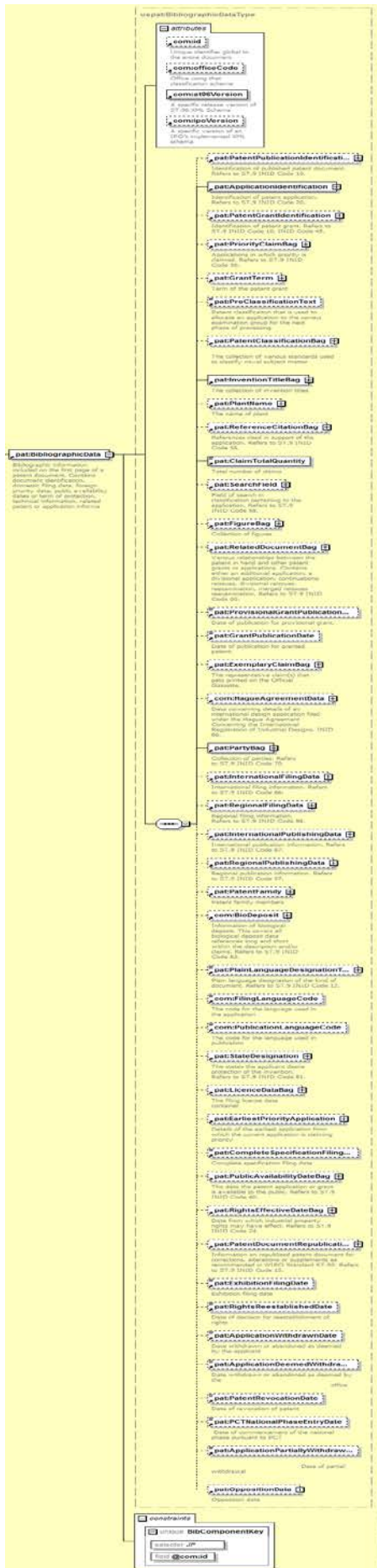
5. In the existing BibliographicData\_V2\_0.xsd file, declare new IPO namespace and import the new schema file created above USBibliographicData\_V1\_0.xsd. Each IPO is expected to version the modified components separate from WIPO because this is the Office specific implementation. "V1\_0" is shown in the following example because this is the first Office modified component.

```
<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:pat="http://www.wipo.int/standards/XMLSchema/ST96/Patent"
xmlns:com="http://www.wipo.int/standards/XMLSchema/ST96/Common"
xmlns:uspat="urn:us:gov:doc:uspto:patent"
targetNamespace="http://www.wipo.int/standards/XMLSchema/ST96/Patent"
elementFormDefault="qualified" attributeFormDefault="qualified"
version="V1_0">
<xsd:import namespace="urn:us:gov:doc:uspto:patent"
schemaLocation="USBibliographicData_V1_0.xsd"/>
```

6. Update the existing element BibliographicData type namespace reference based on the corresponding Type above.

```
<xsd:element name="BibliographicData" type="uspat:BibliographicDataType">
  <xsd:annotation>
    <xsd:documentation>Bibliographic information included on the
first page of a patent
document. Contains document identification, domestic
filing data, foreign priority
data, public availability dates or term of protection,
technical information,
related patent or application
informa</xsd:documentation>
  </xsd:annotation>
  <xsd:unique name="BibComponentKey">
    <xsd:selector xpath=".*"/>
    <xsd:field xpath="@com:id"/>
  </xsd:unique>
</xsd:element>
```

7. The final content model design is shown below:



**APPENDIX B - EXAMPLES OF ST.96 COMPATIBLE SCHEMAS**

The following examples are listed to show how developers can reuse ST.96 Components in their implementation schemas to make them compatible with ST.96 Schemas. Note that the target Namespace for compatible schemas MUST be the IPO namespace.

1. COMPLEX TYPES SCHEMAS

1.1 *Design-Stage Schemas - Modular Schemas: Add a new element*

**Trademark Sample Business Case: Add new InternationalRegistrationExpirationDate date element to TrademarkApplication Structure.**

1. Create a folder called "USTRademark" at the same level as "Patent", "Trademark", "Common".
2. Under USTRademark create new subfolder called "Document".
3. Create a new W3C schema file called InternationalRegistrationExpirationDate.xsd in the folder "USTRademark".
4. Set the target Namespace to the IPO's namespace and add the new namespace xmlns:xxtmk, where xx is an ST.3 Code. Define new element InternationalRegistrationExpirationDate as indicated below.

```
<?xml version="1.0" encoding="UTF-8"?>
<xsd:schema xmlns:com="http://www.wipo.int/standards/XMLSchema/ST96/Common"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:ustmk="urn:us:gov:doc:uspto:trademark"
targetNamespace="urn:us:gov:doc:uspto:trademark"
elementFormDefault="qualified" attributeFormDefault="qualified"
version="V1_0">
  <xsd:import
namespace="http://www.wipo.int/standards/XMLSchema/ST96/Common"
schemaLocation=" ../Common/DateType.xsd"/>
  <xsd:element name="InternationalRegistrationExpirationDate"
type="com:DateType">
    <xsd:annotation>
      <xsd:documentation>International Registration Expiration
Date</xsd:documentation>
    </xsd:annotation>
  </xsd:element>
</xsd:schema>
```

5. Create a new W3C schema file called PreviousRegistrationType.xsd in the folder "USTRademark".
6. Set the target Namespace to the IPO's namespace and add the new namespace xmlns:xxtmk, where xx is an ST.3 Code. Define new element PreviousRegistrationType as indicated below.

```
<?xml version="1.0" encoding="UTF-8"?>
<xsd:schema xmlns:com="http://www.wipo.int/standards/XMLSchema/ST96/Common"
xmlns:tmk="http://www.wipo.int/standards/XMLSchema/ST96/Trademark"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:ustmk="urn:us:gov:doc:uspto:trademark"
targetNamespace="urn:us:gov:doc:uspto:trademark"
elementFormDefault="qualified" attributeFormDefault="qualified"
version="V1_0">
  <xsd:import
namespace="http://www.wipo.int/standards/XMLSchema/ST96/Common"
schemaLocation=" ../Common/RegistrationOfficeCode.xsd"/>
  <xsd:import
namespace="http://www.wipo.int/standards/XMLSchema/ST96/Trademark"
schemaLocation=" ../Trademark/PreviousRegistrationNumber.xsd"/>
  <xsd:import
namespace="http://www.wipo.int/standards/XMLSchema/ST96/Trademark"
schemaLocation=" ../Trademark/PreviousRegistrationDate.xsd"/>
  <xsd:import
namespace="http://www.wipo.int/standards/XMLSchema/ST96/Trademark"
schemaLocation=" ../Trademark/DomesticNotificationDate.xsd"/>
  <xsd:import
namespace="http://www.wipo.int/standards/XMLSchema/ST96/Trademark"
schemaLocation=" ../Trademark/AutomaticProtectionDate.xsd"/>
```



```
<xsd:import
namespace="http://www.wipo.int/standards/XMLSchema/ST96/Trademark"
schemaLocation=" ../Trademark/PreviousRegistrationBasisCategory.xsd"/>
  <xsd:import
namespace="http://www.wipo.int/standards/XMLSchema/ST96/Trademark"
schemaLocation=" ../Trademark/PreviousRegistrationCancellationDate.xsd"/>
    <xsd:import
namespace="http://www.wipo.int/standards/XMLSchema/ST96/Trademark"
schemaLocation=" ../Trademark/PreviousRegistrationPriorityDate.xsd"/>
      <xsd:import
namespace="http://www.wipo.int/standards/XMLSchema/ST96/Trademark"
schemaLocation=" ../Trademark/PreviousRegistrationStatusBag.xsd"/>
        <xsd:import
namespace="http://www.wipo.int/standards/XMLSchema/ST96/Trademark"
schemaLocation=" ../Trademark/PriorityClaimIndicator.xsd"/>
          <xsd:import
namespace="http://www.wipo.int/standards/XMLSchema/ST96/Trademark"
schemaLocation=" ../Trademark/FirstRefusalIssueIndicator.xsd"/>
            <xsd:include
schemaLocation="InternationalRegistrationExpirationDate.xsd"/>
              <xsd:import
namespace="http://www.wipo.int/standards/XMLSchema/ST96/Common"
schemaLocation=" ../Common/CommentText.xsd"/>
                <xsd:complexType name="PreviousRegistrationType">
                  <xsd:sequence>
                    <xsd:element ref="com:RegistrationOfficeCode" minOccurs="0"/>
                    <xsd:element ref="tmk:PreviousRegistrationNumber"/>
                    <xsd:element ref="tmk:PreviousRegistrationDate"
minOccurs="0"/>
                    <xsd:element ref="tmk:DomesticNotificationDate"
minOccurs="0"/>
                    <xsd:element ref="tmk:AutomaticProtectionDate"
minOccurs="0"/>
                    <xsd:element ref="tmk:PreviousRegistrationBasisCategory"
minOccurs="0"/>
                    <xsd:element ref="tmk:PreviousRegistrationCancellationDate"
minOccurs="0"/>
                    <xsd:element ref="tmk:PreviousRegistrationPriorityDate"
minOccurs="0"/>
                    <xsd:element ref="tmk:PreviousRegistrationStatusBag"
minOccurs="0"/>
                    <xsd:element ref="tmk:PriorityClaimIndicator" minOccurs="0"/>
                    <xsd:element ref="tmk:FirstRefusalIssueIndicator"
minOccurs="0"/>
                    <xsd:element
ref="ustmk:InternationalRegistrationExpirationDate" minOccurs="0"/>
                    <xsd:element ref="com:CommentText" minOccurs="0"/>
                  </xsd:sequence>
                </xsd:complexType>
            </xsd:schema>
```

7. Create a new W3C schema file called PreviousRegistration.xsd in the folder "USTrademark".

8. Set the target Namespace to the IPO's namespace and add the new namespace xmlns:xxtmk, where xx is an ST.3 Code. Element PreviousRegistration refers to the PreviousRegistrationType defined above.

```
<?xml version="1.0" encoding="UTF-8"?>
<xsd:schema xmlns:com="http://www.wipo.int/standards/XMLSchema/ST96/Common"
xmlns:tmk="http://www.wipo.int/standards/XMLSchema/ST96/Trademark"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:ustmk="urn:us:gov:doc:uspto:trademark"
targetNamespace="urn:us:gov:doc:uspto:trademark"
elementFormDefault="qualified" attributeFormDefault="qualified"
version="V1_0">
  <xsd:include schemaLocation="PreviousRegistrationType.xsd"/>
  <xsd:element name="PreviousRegistration"
type="ustmk:PreviousRegistrationType">
    <xsd:annotation>
      <xsd:documentation>Data regarding a previous
registration</xsd:documentation>
    </xsd:annotation>
  </xsd:element>
</xsd:schema>
```

9. Create a new W3C schema file called PreviousRegistrationBagType.xsd in the folder "USTRademark".
10. Set the target Namespace to the IPO's namespace and add the new namespace xmlns:xxtmk, where xx is an ST.3 Code. Define new complex type PreviousRegistrationBagType as indicated below.

```
<?xml version="1.0" encoding="UTF-8"?>
<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:ustmk="urn:us:gov:doc:uspto:trademark"
targetNamespace="urn:us:gov:doc:uspto:trademark"
elementFormDefault="qualified" attributeFormDefault="qualified"
version="V1_0">
  <xsd:include schemaLocation="PreviousRegistration.xsd"/>
  <xsd:complexType name="PreviousRegistrationBagType">
    <xsd:sequence>
      <xsd:element ref="ustmk:PreviousRegistration"
maxOccurs="unbounded"/>
    </xsd:sequence>
  </xsd:complexType>
</xsd:schema>
```

11. Create a new W3C schema file called PreviousRegistrationBag.xsd in the folder "USTRademark".
12. Set the target Namespace to the IPO's namespace and add the new namespace xmlns:xxtmk, where xx is an ST.3 Code. Element PreviousRegistrationBag refers to the PreviousRegistrationBagType defined above.

```
<?xml version="1.0" encoding="UTF-8"?>
<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:ustmk="urn:us:gov:doc:uspto:trademark"
targetNamespace="urn:us:gov:doc:uspto:trademark"
elementFormDefault="qualified" attributeFormDefault="qualified"
version="V1_0">
  <xsd:include schemaLocation="PreviousRegistrationBagType.xsd"/>
  <xsd:element name="PreviousRegistrationBag"
type="ustmk:PreviousRegistrationBagType">
    <xsd:annotation>
      <xsd:documentation>Collection of previous
registrations</xsd:documentation>
    </xsd:annotation>
  </xsd:element>
</xsd:schema>
```

13. Create a new W3C schema file called TrademarkType.xsd in the folder "USTRademark".
14. Set the target Namespace to the IPO's namespace and add the new namespace xmlns:xxtmk, where xx is an ST.3 Code. Define new complex type TrademarkType as indicated below.

```
<?xml version="1.0" encoding="UTF-8"?>
<xsd:schema xmlns:com="http://www.wipo.int/standards/XMLSchema/ST96/Common"
xmlns:tmk="http://www.wipo.int/standards/XMLSchema/ST96/Trademark"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:ustmk="urn:us:gov:doc:uspto:trademark"
targetNamespace="urn:us:gov:doc:uspto:trademark"
elementFormDefault="qualified" attributeFormDefault="qualified"
version="V1_0">
  <xsd:import
namespace="http://www.wipo.int/standards/XMLSchema/ST96/Common"
schemaLocation=" ../Common/RegistrationOfficeCode.xsd" />
  <xsd:import
namespace="http://www.wipo.int/standards/XMLSchema/ST96/Common"
schemaLocation=" ../Common/ReceivingOfficeCode.xsd" />
  <xsd:import
namespace="http://www.wipo.int/standards/XMLSchema/ST96/Common"
schemaLocation=" ../Common/ReceivingOfficeDate.xsd" />
  <xsd:import
namespace="http://www.wipo.int/standards/XMLSchema/ST96/Common"
schemaLocation=" ../Common/ApplicationNumber.xsd" />
  <xsd:import
namespace="http://www.wipo.int/standards/XMLSchema/ST96/Common"
schemaLocation=" ../Common/RegistrationNumber.xsd" />
  <xsd:import
namespace="http://www.wipo.int/standards/XMLSchema/ST96/Trademark"
schemaLocation=" ../Trademark/ApplicationDate.xsd" />
  <xsd:import
namespace="http://www.wipo.int/standards/XMLSchema/ST96/Trademark"
schemaLocation=" ../Trademark/ApplicationDateTime.xsd" />
  <xsd:import
namespace="http://www.wipo.int/standards/XMLSchema/ST96/Common"
schemaLocation=" ../Common/RegistrationDate.xsd" />
  <xsd:import
namespace="http://www.wipo.int/standards/XMLSchema/ST96/Trademark"
schemaLocation=" ../Trademark/OtherDate.xsd" />
  <xsd:import
namespace="http://www.wipo.int/standards/XMLSchema/ST96/Common"
schemaLocation=" ../Common/FilingPlace.xsd" />
  <xsd:import
namespace="http://www.wipo.int/standards/XMLSchema/ST96/Common"
schemaLocation=" ../Common/ApplicantFileReference.xsd" />
  <xsd:import
namespace="http://www.wipo.int/standards/XMLSchema/ST96/Trademark"
schemaLocation=" ../Trademark/ApplicantSideCaseIdentifier.xsd" />
  <xsd:import
namespace="http://www.wipo.int/standards/XMLSchema/ST96/Trademark"
schemaLocation=" ../Trademark/ApplicationLanguageCode.xsd" />
  <xsd:import
namespace="http://www.wipo.int/standards/XMLSchema/ST96/Common"
schemaLocation=" ../Common/SecondLanguageCode.xsd" />
  <xsd:import
namespace="http://www.wipo.int/standards/XMLSchema/ST96/Common"
schemaLocation=" ../Common/CorrespondenceLanguageCode.xsd" />
  <xsd:import
namespace="http://www.wipo.int/standards/XMLSchema/ST96/Trademark"
schemaLocation=" ../Trademark/ExaminedApplicationNumber.xsd" />
  <xsd:import
namespace="http://www.wipo.int/standards/XMLSchema/ST96/Common"
schemaLocation=" ../Common/ExpiryDate.xsd" />
  <xsd:import
namespace="http://www.wipo.int/standards/XMLSchema/ST96/Trademark"
schemaLocation=" ../Trademark/TerminationDate.xsd" />
  <xsd:import
namespace="http://www.wipo.int/standards/XMLSchema/ST96/Trademark"
schemaLocation=" ../Trademark/MarkCurrentStatusCode.xsd" />
  <xsd:import
namespace="http://www.wipo.int/standards/XMLSchema/ST96/Trademark"
schemaLocation=" ../Trademark/MarkCurrentStatusDate.xsd" />
  <xsd:import
namespace="http://www.wipo.int/standards/XMLSchema/ST96/Trademark"
schemaLocation=" ../Trademark/DesignatedCountryBag.xsd" />
  <xsd:include schemaLocation="PreviousRegistrationBag.xsd" />
```

```
<xsd:import
namespace="http://www.wipo.int/standards/XMLSchema/ST96/Trademark"
schemaLocation=" ../Trademark/AssociatedMark.xsd" />
<xsd:import
namespace="http://www.wipo.int/standards/XMLSchema/ST96/Trademark"
schemaLocation=" ../Trademark/DivisionalApplicationBag.xsd" />
<xsd:import
namespace="http://www.wipo.int/standards/XMLSchema/ST96/Trademark"
schemaLocation=" ../Trademark/ConversionApplicationBag.xsd" />
<xsd:import
namespace="http://www.wipo.int/standards/XMLSchema/ST96/Trademark"
schemaLocation=" ../Trademark/BasicRegistrationApplicationBag.xsd" />
<xsd:import
namespace="http://www.wipo.int/standards/XMLSchema/ST96/Trademark"
schemaLocation=" ../Trademark/InternationalMarkIdentifierBag.xsd" />
<xsd:import
namespace="http://www.wipo.int/standards/XMLSchema/ST96/Trademark"
schemaLocation=" ../Trademark/MarkCategory.xsd" />
<xsd:import
namespace="http://www.wipo.int/standards/XMLSchema/ST96/Trademark"
schemaLocation=" ../Trademark/MarkRepresentation.xsd" />
<xsd:import
namespace="http://www.wipo.int/standards/XMLSchema/ST96/Trademark"
schemaLocation=" ../Trademark/MarkDisclaimerBag.xsd" />
<xsd:import
namespace="http://www.wipo.int/standards/XMLSchema/ST96/Trademark"
schemaLocation=" ../Trademark/NonUseCancelledIndicator.xsd" />
<xsd:import
namespace="http://www.wipo.int/standards/XMLSchema/ST96/Trademark"
schemaLocation=" ../Trademark/SpecialCircumstanceText.xsd" />
<xsd:import
namespace="http://www.wipo.int/standards/XMLSchema/ST96/Trademark"
schemaLocation=" ../Trademark/MarkSeriesTotalQuantity.xsd" />
<xsd:import
namespace="http://www.wipo.int/standards/XMLSchema/ST96/Trademark"
schemaLocation=" ../Trademark/TradeDistinctivenessIndicator.xsd" />
<xsd:import
namespace="http://www.wipo.int/standards/XMLSchema/ST96/Trademark"
schemaLocation=" ../Trademark/TradeDistinctivenessText.xsd" />
<xsd:import
namespace="http://www.wipo.int/standards/XMLSchema/ST96/Trademark"
schemaLocation=" ../Trademark/UseRight.xsd" />
<xsd:import
namespace="http://www.wipo.int/standards/XMLSchema/ST96/Trademark"
schemaLocation=" ../Trademark/AsItIsProtectionIndicator.xsd" />
<xsd:import
namespace="http://www.wipo.int/standards/XMLSchema/ST96/Trademark"
schemaLocation=" ../Trademark/UseRequirementText.xsd" />
<xsd:import
namespace="http://www.wipo.int/standards/XMLSchema/ST96/Trademark"
schemaLocation=" ../Trademark/UseLimitationText.xsd" />
<xsd:import
namespace="http://www.wipo.int/standards/XMLSchema/ST96/Common"
schemaLocation=" ../Common/CommentText.xsd" />
<xsd:import
namespace="http://www.wipo.int/standards/XMLSchema/ST96/Trademark"
schemaLocation=" ../Trademark/OppositionPeriodStartDate.xsd" />
<xsd:import
namespace="http://www.wipo.int/standards/XMLSchema/ST96/Trademark"
schemaLocation=" ../Trademark/OppositionPeriodEndDate.xsd" />
<xsd:import
namespace="http://www.wipo.int/standards/XMLSchema/ST96/Trademark"
schemaLocation=" ../Trademark/GoodsServicesBag.xsd" />
<xsd:import
namespace="http://www.wipo.int/standards/XMLSchema/ST96/Trademark"
schemaLocation=" ../Trademark/PriorityBag.xsd" />
<xsd:import
namespace="http://www.wipo.int/standards/XMLSchema/ST96/Common"
schemaLocation=" ../Common/ExhibitionPriorityBag.xsd" />
<xsd:import
namespace="http://www.wipo.int/standards/XMLSchema/ST96/Trademark"
schemaLocation=" ../Trademark/SeniorityBag.xsd" />
```

```
<xsd:import
namespace="http://www.wipo.int/standards/XMLSchema/ST96/Trademark"
schemaLocation=" ../Trademark/PublicationBag.xsd" />
<xsd:import
namespace="http://www.wipo.int/standards/XMLSchema/ST96/Trademark"
schemaLocation=" ../Trademark/ApplicantBag.xsd" />
<xsd:import
namespace="http://www.wipo.int/standards/XMLSchema/ST96/Common"
schemaLocation=" ../Common/RepresentativeBag.xsd" />
<xsd:import
namespace="http://www.wipo.int/standards/XMLSchema/ST96/Trademark"
schemaLocation=" ../Trademark/NationalRepresentative.xsd" />
<xsd:import
namespace="http://www.wipo.int/standards/XMLSchema/ST96/Trademark"
schemaLocation=" ../Trademark/NationalCorrespondent.xsd" />
<xsd:import
namespace="http://www.wipo.int/standards/XMLSchema/ST96/Trademark"
schemaLocation=" ../Trademark/RecordAttorney.xsd" />
<xsd:import
namespace="http://www.wipo.int/standards/XMLSchema/ST96/Common"
schemaLocation=" ../Common/StaffBag.xsd" />
<xsd:import
namespace="http://www.wipo.int/standards/XMLSchema/ST96/Common"
schemaLocation=" ../Common/Authorization.xsd" />
<xsd:import
namespace="http://www.wipo.int/standards/XMLSchema/ST96/Trademark"
schemaLocation=" ../Trademark/OppositionBag.xsd" />
<xsd:import
namespace="http://www.wipo.int/standards/XMLSchema/ST96/Trademark"
schemaLocation=" ../Trademark/MarkRecordBag.xsd" />
<xsd:import
namespace="http://www.wipo.int/standards/XMLSchema/ST96/Trademark"
schemaLocation=" ../Trademark/MarkEventBag.xsd" />
<xsd:import
namespace="http://www.wipo.int/standards/XMLSchema/ST96/Trademark"
schemaLocation=" ../Trademark/TrademarkDocumentBag.xsd" />
<xsd:import
namespace="http://www.wipo.int/standards/XMLSchema/ST96/Trademark"
schemaLocation=" ../Trademark/MarkUserBag.xsd" />
<xsd:import
namespace="http://www.wipo.int/standards/XMLSchema/ST96/Trademark"
schemaLocation=" ../Trademark/MarkFeatureDescription.xsd" />
<xsd:import
namespace="http://www.wipo.int/standards/XMLSchema/ST96/Common"
schemaLocation=" ../Common/Version.xsd" />
<xsd:import
namespace="http://www.wipo.int/standards/XMLSchema/ST96/Trademark"
schemaLocation=" ../Trademark/AssignmentBag.xsd" />
<xsd:import
namespace="http://www.wipo.int/standards/XMLSchema/ST96/Trademark"
schemaLocation=" ../Trademark/LegalProceedingsBag.xsd" />
<xsd:import
namespace="http://www.wipo.int/standards/XMLSchema/ST96/Trademark"
schemaLocation=" ../Trademark/NationalTrademarkInformation.xsd" />
<xsd:import
namespace="http://www.wipo.int/standards/XMLSchema/ST96/Trademark"
schemaLocation=" ../Trademark/NationalGoodsServices.xsd" />
<xsd:import
namespace="http://www.wipo.int/standards/XMLSchema/ST96/Trademark"
schemaLocation=" ../Trademark/NationalFilingBasis.xsd" />
<xsd:import
namespace="http://www.wipo.int/standards/XMLSchema/ST96/Trademark"
schemaLocation=" ../Trademark/NationalCaseLocation.xsd" />
<xsd:import
namespace="http://www.wipo.int/standards/XMLSchema/ST96/Common"
schemaLocation=" ../Common/CorrespondenceAddress.xsd" />
<xsd:import
namespace="http://www.wipo.int/standards/XMLSchema/ST96/Common"
schemaLocation=" ../Common/operationCategory.xsd" />
<xsd:complexType name="TrademarkType">
<xsd:sequence>
<xsd:element ref="com:RegistrationOfficeCode" minOccurs="0" />
```

```
<xsd:element ref="com:ReceivingOfficeCode" minOccurs="0"/>
<xsd:element ref="com:ReceivingOfficeDate" minOccurs="0"/>
<xsd:choice maxOccurs="unbounded">
  <xsd:element ref="com:ApplicationNumber"/>
  <xsd:element ref="com:RegistrationNumber"/>
</xsd:choice>
<xsd:choice minOccurs="0">
  <xsd:element ref="tmk:ApplicationDate"/>
  <xsd:element ref="tmk:ApplicationDateTime"/>
</xsd:choice>
<xsd:element ref="com:RegistrationDate" minOccurs="0"/>
<xsd:element ref="tmk:OtherDate" minOccurs="0"
maxOccurs="unbounded" />
<xsd:element ref="com:FilingPlace" minOccurs="0"/>
<xsd:element ref="com:ApplicantFileReference" minOccurs="0"/>
<xsd:element ref="tmk:ApplicantSideCaseIdentifier"
minOccurs="0"/>
<xsd:element ref="tmk:ApplicationLanguageCode"
minOccurs="0"/>
<xsd:element ref="com:SecondLanguageCode" minOccurs="0"/>
<xsd:element ref="com:CorrespondenceLanguageCode"
minOccurs="0"/>
<xsd:element ref="tmk:ExaminedApplicationNumber"
minOccurs="0"/>
<xsd:element ref="com:ExpiryDate" minOccurs="0"/>
<xsd:element ref="tmk:TerminationDate" minOccurs="0"/>
<xsd:element ref="tmk:MarkCurrentStatusCode" minOccurs="0"/>
<xsd:element ref="tmk:MarkCurrentStatusDate" minOccurs="0"/>
<xsd:element ref="tmk:DesignatedCountryBag" minOccurs="0"/>
<xsd:element ref="ustmk:PreviousRegistrationBag"
minOccurs="0"/>
<xsd:element ref="tmk:AssociatedMark" minOccurs="0"/>
<xsd:element ref="tmk:DivisionalApplicationBag"
minOccurs="0"/>
<xsd:element ref="tmk:ConversionApplicationBag"
minOccurs="0"/>
<xsd:element ref="tmk:BasicRegistrationApplicationBag"
minOccurs="0"/>
<xsd:element ref="tmk:InternationalMarkIdentifierBag"
minOccurs="0" maxOccurs="unbounded" />
<xsd:element ref="tmk:MarkCategory" minOccurs="0"
maxOccurs="unbounded" />
<xsd:element ref="tmk:MarkRepresentation" minOccurs="0"/>
<xsd:element ref="tmk:MarkDisclaimerBag" minOccurs="0"/>
<xsd:element ref="tmk:NonUseCancelledIndicator"
minOccurs="0"/>
<xsd:element ref="tmk:SpecialCircumstanceText"
minOccurs="0"/>
<xsd:element ref="tmk:MarkSeriesTotalQuantity"
minOccurs="0"/>
<xsd:element ref="tmk:TradeDistinctivenessIndicator"
minOccurs="0"/>
<xsd:element ref="tmk:TradeDistinctivenessText"
minOccurs="0"/>
<xsd:element ref="tmk:UseRight" minOccurs="0"/>
<xsd:element ref="tmk:AsItIsProtectionIndicator"
minOccurs="0"/>
<xsd:element ref="tmk:UseRequirementText" minOccurs="0"/>
<xsd:element ref="tmk:UseLimitationText" minOccurs="0"/>
<xsd:element ref="com:CommentText" minOccurs="0"/>
<xsd:element ref="tmk:OppositionPeriodStartDate"
minOccurs="0"/>
<xsd:element ref="tmk:OppositionPeriodEndDate"
minOccurs="0"/>
<xsd:element ref="tmk:GoodsServicesBag" minOccurs="0"/>
<xsd:element ref="tmk:PriorityBag" minOccurs="0"/>
<xsd:element ref="com:ExhibitionPriorityBag" minOccurs="0"/>
<xsd:element ref="tmk:SeniorityBag" minOccurs="0"/>
<xsd:element ref="tmk:PublicationBag" minOccurs="0"/>
<xsd:element ref="tmk:ApplicantBag" minOccurs="0"/>
<xsd:element ref="com:RepresentativeBag" minOccurs="0"/>
<xsd:element ref="tmk:NationalRepresentative" minOccurs="0"/>
```

```

<xsd:element ref="tmk:NationalCorrespondent" minOccurs="0"/>
<xsd:element ref="tmk:RecordAttorney" minOccurs="0"/>
<xsd:element ref="com:StaffBag" minOccurs="0"/>
<xsd:element ref="com:Authorization" minOccurs="0"/>
<xsd:element ref="tmk:OppositionBag" minOccurs="0"/>
<xsd:element ref="tmk:MarkRecordBag" minOccurs="0"/>
<xsd:element ref="tmk:MarkEventBag" minOccurs="0"/>
<xsd:element ref="tmk:TrademarkDocumentBag" minOccurs="0"/>
<xsd:element ref="tmk:MarkUserBag" minOccurs="0"/>
<xsd:element ref="tmk:MarkFeatureDescription" minOccurs="0"/>
<xsd:element ref="com:Version" minOccurs="0"/>
<xsd:element ref="tmk:AssignmentBag" minOccurs="0"/>
<xsd:element ref="tmk:LegalProceedingsBag" minOccurs="0"/>
<xsd:element ref="tmk:NationalTrademarkInformation"
minOccurs="0"/>
<xsd:element ref="tmk:NationalGoodsServices" minOccurs="0"/>
<xsd:element ref="tmk:NationalFilingBasis" minOccurs="0"/>
<xsd:element ref="tmk:NationalCaseLocation" minOccurs="0"/>
<xsd:element ref="com:CorrespondenceAddress" minOccurs="0"/>
</xsd:sequence>
<xsd:attribute ref="com:operationCategory"/>
</xsd:complexType>
</xsd:schema>

```

15. Create a new W3C schema file called Trademark.xsd in the folder "USTrademark".
16. Set the target Namespace to the IPO's namespace and add the new namespace xmlns:xxtmk, where xx is an ST.3 Code. Element Trademark refers to the TrademarkType defined above.

```

<?xml version="1.0" encoding="UTF-8"?>
<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:ustmk="urn:us:gov:doc:uspto:trademark"
targetNamespace="urn:us:gov:doc:uspto:trademark"
elementFormDefault="qualified" attributeFormDefault="qualified"
version="V1_0">
  <xsd:include schemaLocation="TrademarkType.xsd"/>
  <xsd:element name="Trademark" type="ustmk:TrademarkType">
    <xsd:annotation>
      <xsd:documentation>Core data regarding the trade
mark</xsd:documentation>
    </xsd:annotation>
  </xsd:element>
</xsd:schema>

```

17. Create a new W3C schema file called TrademarkBagType.xsd in the folder "USTrademark".
18. Set the target Namespace to the IPO's namespace and add the new namespace xmlns:xxtmk, where xx is an ST.3 Code. Define new complex type TrademarkBagType as indicated below.

```

<?xml version="1.0" encoding="UTF-8"?>
<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:ustmk="urn:us:gov:doc:uspto:trademark"
targetNamespace="urn:us:gov:doc:uspto:trademark"
elementFormDefault="qualified" attributeFormDefault="qualified"
version="V1_0">
  <xsd:include schemaLocation="Trademark.xsd"/>
  <xsd:complexType name="TrademarkBagType">
    <xsd:sequence>
      <xsd:element ref="ustmk:Trademark" maxOccurs="unbounded"/>
    </xsd:sequence>
  </xsd:complexType>
</xsd:schema>

```

19. Create a new W3C schema file called TrademarkBag.xsd in the folder "USTrademark".
20. Set the target Namespace to the IPO's namespace and add the new namespace xmlns:xxtmk, where xx is an ST.3 Code. Element TrademarkBag refers to the TrademarkBagType defined above.

```
<?xml version="1.0" encoding="UTF-8"?>
<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:ustmk="urn:us:gov:doc:uspto:trademark"
targetNamespace="urn:us:gov:doc:uspto:trademark"
elementFormDefault="qualified" attributeFormDefault="qualified"
version="V1_0">
  <xsd:include schemaLocation="TrademarkBagType.xsd" />
  <xsd:element name="TrademarkBag" type="ustmk:TrademarkBagType">
    <xsd:annotation>
      <xsd:documentation>Collection of
trademarks</xsd:documentation>
    </xsd:annotation>
  </xsd:element>
</xsd:schema>
```

21. Create a new W3C schema file called TrademarkApplicationType.xsd in the "USTrademark" -> "Document" folder.
22. Set the target Namespace to the IPO's namespace and add the new namespace xmlns:xtmk, where xx is an ST.3 Code.

```
<?xml version="1.0" encoding="UTF-8"?>
<xsd:schema xmlns:com="http://www.wipo.int/standards/XMLSchema/ST96/Common"
xmlns:tmk="http://www.wipo.int/standards/XMLSchema/ST96/Trademark"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:ustmk="urn:us:gov:doc:uspto:trademark"
targetNamespace="urn:us:gov:doc:uspto:trademark"
elementFormDefault="qualified" attributeFormDefault="qualified"
version="V1_0">
```

23. Import ST.96 Components which are needed to define a new Type TrademarkApplicationType.

```
  <xsd:import
namespace="http://www.wipo.int/standards/XMLSchema/ST96/Common"
schemaLocation="../../../Common/RequestSoftware.xsd"/>
  <xsd:import
namespace="http://www.wipo.int/standards/XMLSchema/ST96/Trademark"
schemaLocation="../../../Trademark/ApplicationFormName.xsd"/>
  <xsd:import
namespace="http://www.wipo.int/standards/XMLSchema/ST96/Trademark"
schemaLocation="../../../Trademark/RequestSearch.xsd"/>
  <xsd:import
namespace="http://www.wipo.int/standards/XMLSchema/ST96/Common"
schemaLocation="../../../Common/RequestExamination.xsd"/>
  <xsd:import
namespace="http://www.wipo.int/standards/XMLSchema/ST96/Common"
schemaLocation="../../../Common/DocumentIncludedBag.xsd"/>
  <xsd:include schemaLocation="../../TrademarkBag.xsd"/>
  <xsd:import
namespace="http://www.wipo.int/standards/XMLSchema/ST96/Common"
schemaLocation="../../../Common/PaymentBag.xsd"/>
  <xsd:import
namespace="http://www.wipo.int/standards/XMLSchema/ST96/Common"
schemaLocation="../../../Common/ReimbursementBag.xsd"/>
  <xsd:import
namespace="http://www.wipo.int/standards/XMLSchema/ST96/Common"
schemaLocation="../../../Common/SignatureBag.xsd"/>
  <xsd:import
namespace="http://www.wipo.int/standards/XMLSchema/ST96/Common"
schemaLocation="../../../Common/st96Version.xsd"/>
  <xsd:import
namespace="http://www.wipo.int/standards/XMLSchema/ST96/Common"
schemaLocation="../../../Common/ipoVersion.xsd"/>
```



24. Create the new Complex Type TrademarkApplicationType which refers to existing elements and the new element called InternationalRegistrationExpirationDate being referred through TrademarkApplicationType -> TrademarkBag -> TrademarkBagType -> Trademark -> TrademarkType -> PreviousRegistrationBag -> PreviousRegistration -> InternationalRegistrationExpirationDate.

```
<xsd:complexType name="TrademarkApplicationType">
  <xsd:sequence>
    <xsd:element ref="com:RequestSoftware" minOccurs="0"/>
    <xsd:element ref="tmk:ApplicationFormName" minOccurs="0"/>
    <xsd:element ref="tmk:RequestSearch" minOccurs="0"/>
    <xsd:element ref="com:RequestExamination" minOccurs="0"/>
    <xsd:element ref="com:DocumentIncludedBag" minOccurs="0"/>
    <xsd:element ref="ustmk:TrademarkBag"/>
    <xsd:element ref="com:PaymentBag" minOccurs="0"/>
    <xsd:element ref="com:ReimbursementBag" minOccurs="0"/>
    <xsd:element ref="com:SignatureBag" minOccurs="0"/>
  </xsd:sequence>
  <xsd:attribute ref="com:st96Version" use="required"/>
  <xsd:attribute ref="com:ipoVersion"/>
</xsd:complexType>
</xsd:schema>
```

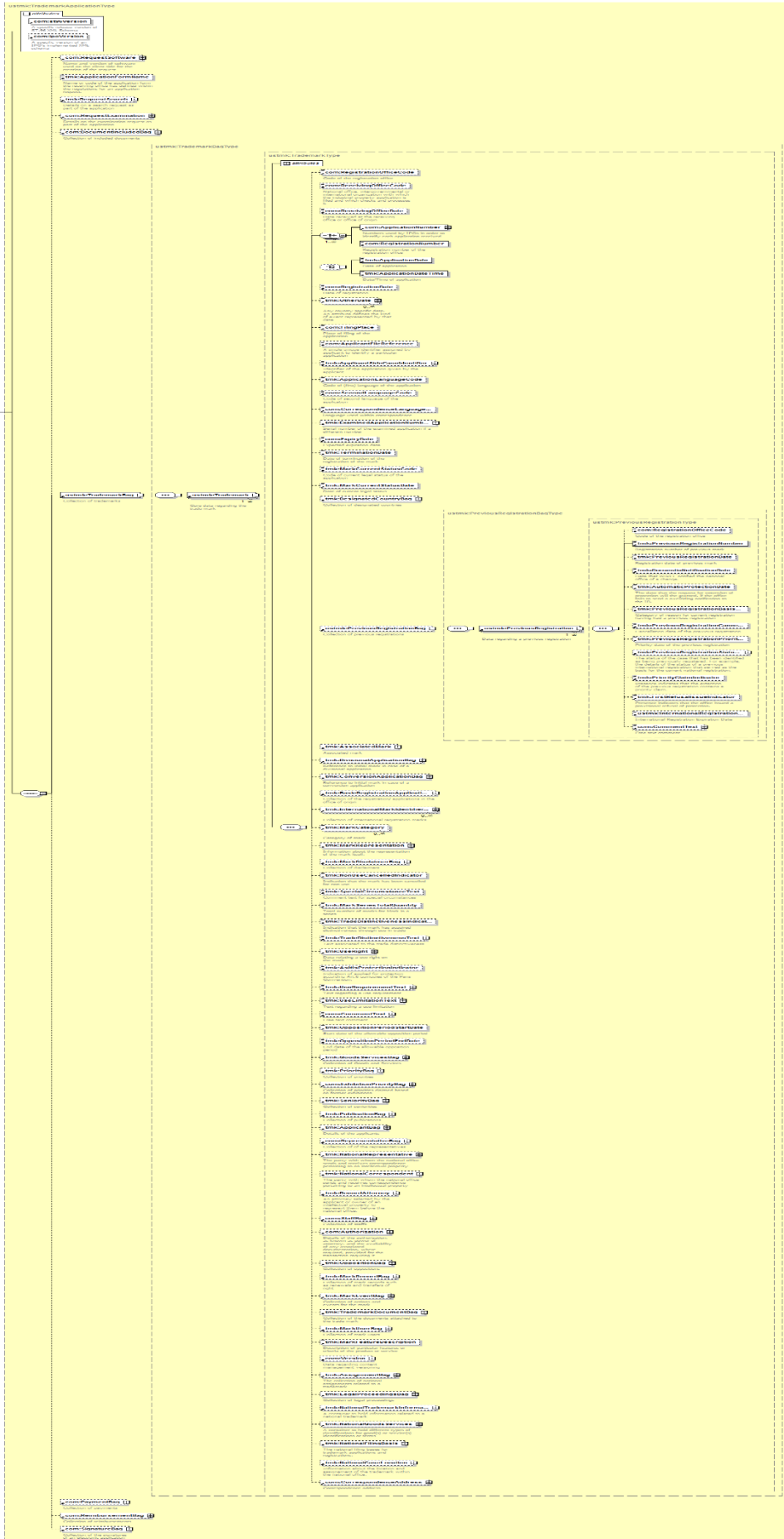
25. Create a new W3C schema file called TrademarkApplication.xsd in the folder "USTrademark"-> "Document" folder.
26. Set the target Namespace to the IPO's namespace and add the new namespace xmlns:ustmk, where xx is an ST.3 Code.

```
<?xml version="1.0" encoding="UTF-8"?>
<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:ustmk="urn:us:gov:doc:uspto:trademark"
targetNamespace="urn:us:gov:doc:uspto:trademark"
elementFormDefault="qualified" attributeFormDefault="qualified"
version="V1_0">
```

27. Declare the new element TrademarkApplication which refers to TrademarkApplicationType defined above.

```
<xsd:include schemaLocation="TrademarkApplicationType.xsd"/>
<xsd:element name="TrademarkApplication"
type="ustmk:TrademarkApplicationType">
  <xsd:annotation>
    <xsd:documentation>Details of a trademark
application</xsd:documentation>
  </xsd:annotation>
</xsd:element>
</xsd:schema>
```

28. The final content model design is shown below.



## 1.2 Design-Stage Schemas - Modular Schemas: Remove existing element and add a new element

In order to add a new element and remove an existing one, developers SHOULD follow the following steps:

### Trademark Sample Business Case: Remove the RequestSoftware element and add the ApplicationCategory element to the TrademarkApplication Structure:

1. Create a folder called "USTrademark" at the same level as "Patent", "Trademark", "Common".
2. Under USTrademark create new subfolder called "Document".
3. Create a new W3C schema file called ApplicationCategoryType.xsd in the folder "USTrademark".
4. Set the target Namespace to the IPO's namespace and add the new namespace xmlns:xxtmk, where xx is an ST.3 Code. Define ApplicationCategoryType as indicated below.

```
<?xml version="1.0" encoding="UTF-8"?>
<xsd:schema xmlns:com="http://www.wipo.int/standards/XMLSchema/ST96/Common"
xmlns:tmk="http://www.wipo.int/standards/XMLSchema/ST96/Trademark"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:ustmk="urn:us:gov:doc:uspto:trademark"
targetNamespace="urn:us:gov:doc:uspto:trademark"
elementFormDefault="qualified" attributeFormDefault="qualified"
version="V1_0">
  <xsd:annotation>
    <xsd:documentation>
      Last update: 2013-01-01
      Contact Point: xml.standards@uspto.gov
      ReleaseNotesURL:
    </xsd:documentation>
  </xsd:annotation>
  <xsd:simpleType name="ApplicationCategoryType">
    <xsd:restriction base="xsd:token">
      <xsd:enumeration value="domestic">
        <xsd:annotation>
          <xsd:documentation>Domestic TM
        </xsd:documentation>
        </xsd:annotation>
      </xsd:enumeration>
      <xsd:enumeration value="international">
        <xsd:annotation>
          <xsd:documentation>International TM
        </xsd:documentation>
        </xsd:annotation>
      </xsd:enumeration>
    </xsd:restriction>
  </xsd:simpleType>
</xsd:schema>
```

5. Create a new W3C schema file called ApplicationCategory.xsd in the folder "USTrademark".
6. Set the target Namespace to the IPO's namespace and add the new namespace xmlns:xxtmk, where xx is an ST.3 Code. ApplicationCategory element should refer to the ApplicationCategoryType simple type defined above.

```
<?xml version="1.0" encoding="UTF-8"?>
<xsd:schema xmlns:com="http://www.wipo.int/standards/XMLSchema/ST96/Common"
xmlns:tmk="http://www.wipo.int/standards/XMLSchema/ST96/Trademark"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:ustmk="urn:us:gov:doc:uspto:trademark"
targetNamespace="urn:us:gov:doc:uspto:trademark"
elementFormDefault="qualified" attributeFormDefault="qualified"
version="V1_0">
  <xsd:annotation>
    <xsd:documentation>
      Last update: 2013-01-01
      Contact Point: xml.standards@uspto.gov
      ReleaseNotesURL:
    </xsd:documentation>
  </xsd:annotation>
  <xsd:include schemaLocation="ApplicationCategoryType.xsd"/>
  <xsd:element name="ApplicationCategory"
type="ustmk:ApplicationCategoryType">
```

```
<xsd:annotation>
  <xsd:documentation>
Describes type of Trademark application
</xsd:documentation>
  </xsd:annotation>
</xsd:element>
</xsd:schema>
```

7. Create a new W3C schema file called TrademarkApplicationType.xsd in "USTrademark" -> "Document" folder.
8. Set the target Namespace to the IPO's namespace and add the new namespace xmlns:xtrmk, where xx is an ST.3 Code.

```
<?xml version="1.0" encoding="UTF-8"?>
<xsd:schema xmlns:com="http://www.wipo.int/standards/XMLSchema/ST96/Common"
xmlns:tmk="http://www.wipo.int/standards/XMLSchema/ST96/Trademark"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:ustmk="urn:us:gov:doc:uspto:trademark"
targetNamespace="urn:us:gov:doc:uspto:trademark"
elementFormDefault="qualified" attributeFormDefault="qualified"
version="V1_0">
```

9. Import the ST.96 Components which are needed to define a new Type TrademarkApplicationType. Do not import RequestSoftware.xsd because that element needs to be removed.

```
<xsd:include schemaLocation=" ../ApplicationCategory.xsd" />
<xsd:import
<xsd:import
namespace="http://www.wipo.int/standards/XMLSchema/ST96/Trademark"
schemaLocation=" ../ ../Trademark/RequestSearch.xsd" />
  <xsd:import
namespace="http://www.wipo.int/standards/XMLSchema/ST96/Common"
schemaLocation=" ../ ../Common/RequestExamination.xsd" />
  <xsd:import
namespace="http://www.wipo.int/standards/XMLSchema/ST96/Common"
schemaLocation=" ../ ../Common/DocumentIncludedBag.xsd" />
  <xsd:import
namespace="http://www.wipo.int/standards/XMLSchema/ST96/Trademark"
schemaLocation=" ../ ../Trademark/TrademarkBag.xsd" />
  <xsd:import
namespace="http://www.wipo.int/standards/XMLSchema/ST96/Common"
schemaLocation=" ../ ../Common/PaymentBag.xsd" />
  <xsd:import
namespace="http://www.wipo.int/standards/XMLSchema/ST96/Common"
schemaLocation=" ../ ../Common/ReimbursementBag.xsd" />
  <xsd:import
namespace="http://www.wipo.int/standards/XMLSchema/ST96/Common"
schemaLocation=" ../ ../Common/SignatureBag.xsd" />
  <xsd:import
namespace="http://www.wipo.int/standards/XMLSchema/ST96/Common"
schemaLocation=" ../ ../Common/st96Version.xsd" />
  <xsd:import
namespace="http://www.wipo.int/standards/XMLSchema/ST96/Common"
schemaLocation=" ../ ../Common/ipoVersion.xsd" />
```

10. Create the new Complex Type `TrademarkApplicationType` which refers to the existing elements except `RequestSoftware` and add the new element called `ApplicationCategory`.

```
<xsd:complexType name="TrademarkApplicationType">
  <xsd:sequence>
    <xsd:element ref="ustmk:ApplicationCategory" minOccurs="0"/>
    <xsd:element ref="tmk:RequestSearch" minOccurs="0"/>
    <xsd:element ref="com:RequestExamination" minOccurs="0"/>
    <xsd:element ref="com:DocumentIncludedBag" minOccurs="0"/>
    <xsd:element ref="tmk:TrademarkBag"/>
    <xsd:element ref="com:PaymentBag" minOccurs="0"/>
    <xsd:element ref="com:ReimbursementBag" minOccurs="0"/>
    <xsd:element ref="com:SignatureBag" minOccurs="0"/>
  </xsd:sequence>
  <xsd:attribute ref="com:st96Version" use="required"/>
  <xsd:attribute ref="com:ipoVersion"/>
</xsd:complexType>
```

11. Create a new W3C schema file called `TrademarkApplication.xsd` in the folder "USTRademark" -> "Document" folder.
12. Set the target Namespace to the IPO's namespace and add the new namespace `xmlns:xxtmk`, where `xx` is an ST.3 Code.

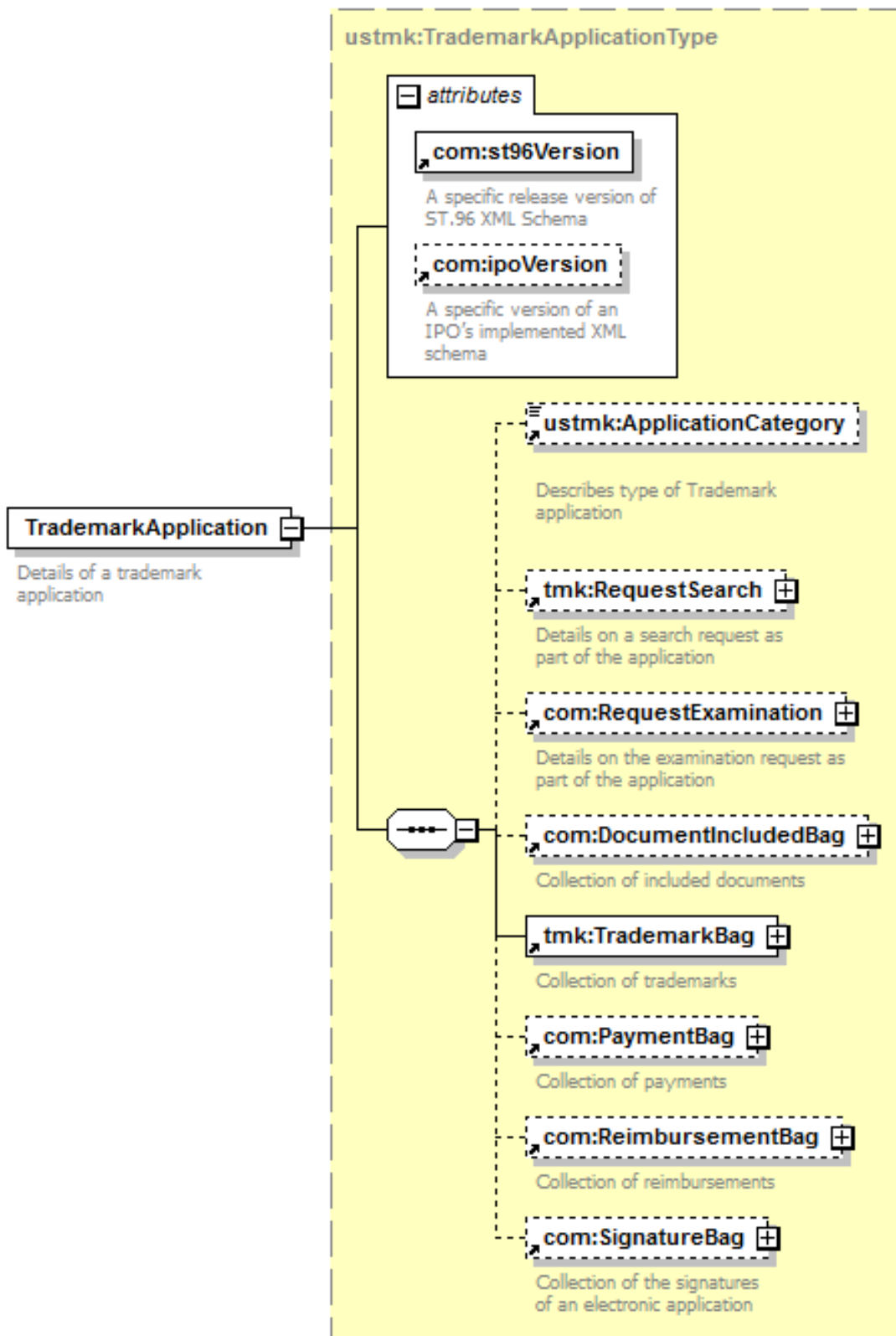
```
<?xml version="1.0" encoding="UTF-8"?>
<xsd:schema xmlns:com="http://www.wipo.int/standards/XMLSchema/ST96/Common"
xmlns:tmk="http://www.wipo.int/standards/XMLSchema/ST96/Trademark"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:ustmk="urn:us:gov:doc:uspto:trademark"
targetNamespace="urn:us:gov:doc:uspto:trademark"
elementFormDefault="qualified" attributeFormDefault="qualified"
version="V1_0">
```

13. Declare the new element `TrademarkApplication` which refers to `TrademarkApplicationType` defined above.

```
  <xsd:include schemaLocation="TrademarkApplicationType.xsd"/>
  <xsd:annotation>
    <xsd:appinfo>
      <com:SchemaCreatedDate>2015-02-20</com:SchemaCreatedDate>
      <com:SchemaLastModifiedDate>2015-03-06</com:SchemaLastModifiedDate>
      <com:SchemaContactPoint>xml.standards@wipo.int</com:SchemaContactPoint>
      <com:SchemaReleaseNoteURL>http://www.wipo.int/standards/XMLSchema/ST96/V2_0/ReleaseNotes.pdf</com:SchemaReleaseNoteURL>
    </xsd:appinfo>
  </xsd:annotation>

  <xsd:element name="TrademarkApplication"
type="ustmk:TrademarkApplicationType">
    <xsd:annotation>
      <xsd:documentation>Details of a trademark
application</xsd:documentation>
    </xsd:annotation>
  </xsd:element>
```

14. The final content model design is shown below:



1.3 Design-Stage Schemas - Flattened Schemas: Add a new element

**Trademark Sample Business Case: Add new InternationalRegistrationExpirationDate date element to TrademarkApplication Structure.**

1. Create a new W3C schema file, USTrademarkApplication\_V1\_0.xsd in the same folder as TrademarkApplication\_V2\_0.xsd.
2. Set the target Namespace to the IPO's namespace and add the new namespace xmlns:xxtmk, where xx is an ST.3 Code.

```
<?xml version="1.0" encoding="UTF-8"?>
<xsd:schema xmlns:com="http://www.wipo.int/standards/XMLSchema/ST96/Common"
xmlns:tmk="http://www.wipo.int/standards/XMLSchema/ST96/Trademark"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:ustmk="urn:us:gov:doc:uspto:trademark"
targetNamespace="urn:us:gov:doc:uspto:trademark"
elementFormDefault="qualified" attributeFormDefault="qualified"
version="V1_0">
```

3. Import ST.96 Components which are needed to define a new InternationalRegistrationExpirationDate element in PreviousRegistration.

```
<xsd:import
namespace="http://www.wipo.int/standards/XMLSchema/ST96/Common"
schemaLocation="Common_V2_0.xsd"/>
<xsd:import
namespace="http://www.wipo.int/standards/XMLSchema/ST96/Trademark"
schemaLocation="TrademarkApplication_V2_0.xsd"/>
```

4. Define InternationalRegistrationExpirationDate as indicated below.

```
<xsd:element name="InternationalRegistrationExpirationDate"
type="com:DateType">
  <xsd:annotation>
    <xsd:documentation>International Registration Expiration
Date</xsd:documentation>
  </xsd:annotation>
</xsd:element>
```

5. Remove PreviousRegistrationType complex type from ST.96 Trademark namespace (TrademarkApplication\_V2\_0.xsd) and declare new complex type PreviousRegistrationType as defined in USTrademarkApplication\_V1\_0.xsd.

```
<xsd:complexType name="PreviousRegistrationType">
  <xsd:sequence>
    <xsd:element ref="com:RegistrationOfficeCode" minOccurs="0"/>
    <xsd:element ref="tmk:PreviousRegistrationNumber"/>
    <xsd:element ref="tmk:PreviousRegistrationDate"
minOccurs="0"/>
    <xsd:element ref="tmk:DomesticNotificationDate"
minOccurs="0"/>
    <xsd:element ref="tmk:AutomaticProtectionDate"
minOccurs="0"/>
    <xsd:element ref="tmk:PreviousRegistrationBasisCategory"
minOccurs="0"/>
    <xsd:element ref="tmk:PreviousRegistrationCancellationDate"
minOccurs="0"/>
    <xsd:element ref="tmk:PreviousRegistrationPriorityDate"
minOccurs="0"/>
    <xsd:element ref="tmk:PreviousRegistrationStatusBag"
minOccurs="0"/>
    <xsd:element ref="tmk:PriorityClaimIndicator" minOccurs="0"/>
    <xsd:element ref="tmk:FirstRefusalIssueIndicator"
minOccurs="0"/>
    <xsd:element
ref="ustmk:InternationalRegistrationExpirationDate" minOccurs="0"/>
    <xsd:element ref="com:CommentText" minOccurs="0"/>
  </xsd:sequence>
</xsd:complexType>
```

6. Remove PreviousRegistration element from ST.96 Trademark namespace (TrademarkApplication\_V2\_0.xsd) and declare new element PreviousRegistration as defined in USTrademarkApplication\_V1\_0.xsd.

```
<xsd:element name="PreviousRegistration"
type="ustmk:PreviousRegistrationType">
  <xsd:annotation>
    <xsd:documentation>Data regarding a previous
registration</xsd:documentation>
  </xsd:annotation>
</xsd:element>
```

7. Remove PreviousRegistrationBagType complex type from ST.96 Trademark namespace (TrademarkApplication\_V2\_0.xsd) and declare new complex type PreviousRegistrationBagType as defined in USTrademarkApplication\_V1\_0.xsd.

```
<xsd:complexType name="PreviousRegistrationBagType">
  <xsd:sequence>
    <xsd:element ref="ustmk:PreviousRegistration"
maxOccurs="unbounded"/>
  </xsd:sequence>
</xsd:complexType>
```

8. Remove PreviousRegistrationBag element from ST.96 Trademark namespace (TrademarkApplication\_V2\_0.xsd) and declare new element PreviousRegistrationBag as defined in USTrademarkApplication\_V1\_0.xsd.

```
<xsd:element name="PreviousRegistrationBag"
type="ustmk:PreviousRegistrationBagType">
  <xsd:annotation>
    <xsd:documentation>Collection of previous
registrations</xsd:documentation>
  </xsd:annotation>
</xsd:element>
```

9. Remove TrademarkType complex type from ST.96 Trademark namespace (TrademarkApplication\_V2\_0.xsd) and declare new complex type TrademarkType as defined in USTrademarkApplication\_V1\_0.xsd.

```
<xsd:complexType name="TrademarkType">
  <xsd:sequence>
    <xsd:element ref="com:RegistrationOfficeCode" minOccurs="0"/>
    <xsd:element ref="com:ReceivingOfficeCode" minOccurs="0"/>
    <xsd:element ref="com:ReceivingOfficeDate" minOccurs="0"/>
    <xsd:choice maxOccurs="unbounded">
      <xsd:element ref="com:ApplicationNumber"/>
      <xsd:element ref="com:RegistrationNumber"/>
    </xsd:choice>
    <xsd:choice minOccurs="0">
      <xsd:element ref="tmk:ApplicationDate"/>
      <xsd:element ref="tmk:ApplicationDateTime"/>
    </xsd:choice>
    <xsd:element ref="com:RegistrationDate" minOccurs="0"/>
    <xsd:element ref="tmk:OtherDate" minOccurs="0"
maxOccurs="unbounded"/>
    <xsd:element ref="com:FilingPlace" minOccurs="0"/>
    <xsd:element ref="com:ApplicantFileReference" minOccurs="0"/>
    <xsd:element ref="tmk:ApplicantSideCaseIdentifier"
minOccurs="0"/>
    <xsd:element ref="tmk:ApplicationLanguageCode"
minOccurs="0"/>
    <xsd:element ref="com:SecondLanguageCode" minOccurs="0"/>
    <xsd:element ref="com:CorrespondenceLanguageCode"
minOccurs="0"/>
    <xsd:element ref="tmk:ExaminedApplicationNumber"
minOccurs="0"/>
    <xsd:element ref="com:ExpiryDate" minOccurs="0"/>
    <xsd:element ref="tmk:TerminationDate" minOccurs="0"/>
    <xsd:element ref="tmk:MarkCurrentStatusCode" minOccurs="0"/>
    <xsd:element ref="tmk:MarkCurrentStatusDate" minOccurs="0"/>
```



```

    <xsd:element ref="tmk:DesignatedCountryBag" minOccurs="0"/>
    <xsd:element ref="ustmk:PreviousRegistrationBag"
minOccurs="0"/>
    <xsd:element ref="tmk:AssociatedMark" minOccurs="0"/>
    <xsd:element ref="tmk:DivisionalApplicationBag"
minOccurs="0"/>
    <xsd:element ref="tmk:ConversionApplicationBag"
minOccurs="0"/>
    <xsd:element ref="tmk:BasicRegistrationApplicationBag"
minOccurs="0"/>
    <xsd:element ref="tmk:InternationalMarkIdentifierBag"
minOccurs="0" maxOccurs="unbounded"/>
    <xsd:element ref="tmk:MarkCategory" minOccurs="0"
maxOccurs="unbounded"/>
    <xsd:element ref="tmk:MarkRepresentation" minOccurs="0"/>
    <xsd:element ref="tmk:MarkDisclaimerBag" minOccurs="0"/>
    <xsd:element ref="tmk:NonUseCancelledIndicator"
minOccurs="0"/>
    <xsd:element ref="tmk:SpecialCircumstanceText"
minOccurs="0"/>
    <xsd:element ref="tmk:MarkSeriesTotalQuantity"
minOccurs="0"/>
    <xsd:element ref="tmk:TradeDistinctivenessIndicator"
minOccurs="0"/>
    <xsd:element ref="tmk:TradeDistinctivenessText"
minOccurs="0"/>
    <xsd:element ref="tmk:UseRight" minOccurs="0"/>
    <xsd:element ref="tmk:AsItIsProtectionIndicator"
minOccurs="0"/>
    <xsd:element ref="tmk:UseRequirementText" minOccurs="0"/>
    <xsd:element ref="tmk:UseLimitationText" minOccurs="0"/>
    <xsd:element ref="com:CommentText" minOccurs="0"/>
    <xsd:element ref="tmk:OppositionPeriodStartDate"
minOccurs="0"/>
    <xsd:element ref="tmk:OppositionPeriodEndDate"
minOccurs="0"/>
    <xsd:element ref="tmk:GoodsServicesBag" minOccurs="0"/>
    <xsd:element ref="tmk:PriorityBag" minOccurs="0"/>
    <xsd:element ref="com:ExhibitionPriorityBag" minOccurs="0"/>
    <xsd:element ref="tmk:SeniorityBag" minOccurs="0"/>
    <xsd:element ref="tmk:PublicationBag" minOccurs="0"/>
    <xsd:element ref="tmk:ApplicantBag" minOccurs="0"/>
    <xsd:element ref="com:RepresentativeBag" minOccurs="0"/>
    <xsd:element ref="tmk:NationalRepresentative" minOccurs="0"/>
    <xsd:element ref="tmk:NationalCorrespondent" minOccurs="0"/>
    <xsd:element ref="tmk:RecordAttorney" minOccurs="0"/>
    <xsd:element ref="com:StaffBag" minOccurs="0"/>
    <xsd:element ref="com:Authorization" minOccurs="0"/>
    <xsd:element ref="tmk:OppositionBag" minOccurs="0"/>
    <xsd:element ref="tmk:MarkRecordBag" minOccurs="0"/>
    <xsd:element ref="tmk:MarkEventBag" minOccurs="0"/>
    <xsd:element ref="tmk:TrademarkDocumentBag" minOccurs="0"/>
    <xsd:element ref="tmk:MarkUserBag" minOccurs="0"/>
    <xsd:element ref="tmk:MarkFeatureDescription" minOccurs="0"/>
    <xsd:element ref="com:Version" minOccurs="0"/>
    <xsd:element ref="tmk:AssignmentBag" minOccurs="0"/>
    <xsd:element ref="tmk:LegalProceedingsBag" minOccurs="0"/>
    <xsd:element ref="tmk:NationalTrademarkInformation"
minOccurs="0"/>
    <xsd:element ref="tmk:NationalGoodsServices" minOccurs="0"/>
    <xsd:element ref="tmk:NationalFilingBasis" minOccurs="0"/>
    <xsd:element ref="tmk:NationalCaseLocation" minOccurs="0"/>
    <xsd:element ref="com:CorrespondenceAddress" minOccurs="0"/>
  </xsd:sequence>
  <xsd:attribute ref="com:operationCategory"/>
</xsd:complexType>

```

10. Remove Trademark element from ST.96 Trademark namespace (TrademarkApplication\_V2\_0.xsd) and declare new element Trademark as defined in USTrademarkApplication\_V1\_0.xsd.

```
<xsd:element name="Trademark" type="ustmk:TrademarkType">
  <xsd:annotation>
    <xsd:documentation>Core data regarding the trade
mark</xsd:documentation>
  </xsd:annotation>
</xsd:element>
```

11. Remove TrademarkBagType complex type from ST.96 Trademark namespace (TrademarkApplication\_V2\_0.xsd) and declare new complex type TrademarkBagType as defined in USTrademarkApplication\_V1\_0.xsd.

```
<xsd:complexType name="TrademarkBagType">
  <xsd:sequence>
    <xsd:element ref="ustmk:Trademark" maxOccurs="unbounded" />
  </xsd:sequence>
</xsd:complexType>
```

12. Remove TrademarkBag element from ST.96 Trademark namespace (TrademarkApplication\_V2\_0.xsd) and declare new element TrademarkBag as defined in USTrademarkApplication\_V1\_0.xsd.

```
<xsd:element name="TrademarkBag" type="ustmk:TrademarkBagType">
  <xsd:annotation>
    <xsd:documentation>Collection of
trademarks</xsd:documentation>
  </xsd:annotation>
</xsd:element>
```

13. Remove TrademarkApplicationType complex type from ST.96 Trademark namespace (TrademarkApplication\_V2\_0.xsd) and declare new complex type TrademarkApplicationType as defined in USTrademarkApplication\_V1\_0.xsd.

```
<xsd:complexType name="TrademarkApplicationType">
  <xsd:sequence>
    <xsd:element ref="com:RequestSoftware" minOccurs="0" />
    <xsd:element ref="tmk:ApplicationFormName" minOccurs="0" />
    <xsd:element ref="tmk:RequestSearch" minOccurs="0" />
    <xsd:element ref="com:RequestExamination" minOccurs="0" />
    <xsd:element ref="com:DocumentIncludedBag" minOccurs="0" />
    <xsd:element ref="ustmk:TrademarkBag" />
    <xsd:element ref="com:PaymentBag" minOccurs="0" />
    <xsd:element ref="com:ReimbursementBag" minOccurs="0" />
    <xsd:element ref="com:SignatureBag" minOccurs="0" />
  </xsd:sequence>
  <xsd:attribute ref="com:st96Version" use="required" />
  <xsd:attribute ref="com:ipoVersion" />
</xsd:complexType>
```

14. Remove TrademarkApplication element from ST.96 Trademark namespace (TrademarkApplication\_V2\_0.xsd) and declare new element TrademarkApplication as defined in USTrademarkApplication\_V1\_0.xsd.

```
<xsd:element name="TrademarkApplication"
type="ustmk:TrademarkApplicationType">
  <xsd:annotation>
    <xsd:documentation>Details of a trademark
application</xsd:documentation>
  </xsd:annotation>
</xsd:element>
```

15. The final content model design is shown below.



1.4 Design-Stage Schemas - Flattened Schemas: Remove existing element and add a new element

In order to add a new element and remove an existing one, developers SHOULD follow the following steps:

**Trademark Sample Business Case: Remove the RequestSoftware element and add the ApplicationCategory element to the TrademarkApplication Structure:**

1. Create a new W3C schema file, USTrademarkApplication\_V1\_0.xsd in the same folder as TrademarkApplication\_V2\_0.xsd.
2. Set the target Namespace to the IPO's namespace and add the new namespace xmlns:xxtmk, where xx is an ST.3 Code.

```
<?xml version="1.0" encoding="UTF-8"?>
<xsd:schema xmlns:com="http://www.wipo.int/standards/XMLSchema/ST96/Common"
xmlns:tmk="http://www.wipo.int/standards/XMLSchema/ST96/Trademark"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:ustmk="urn:us:gov:doc:uspto:trademark"
targetNamespace="urn:us:gov:doc:uspto:trademark"
elementFormDefault="qualified" attributeFormDefault="qualified"
version="V1_0">
```

3. Import ST.96 Components which are needed to define a new Type TrademarkApplicationType.

```
<xsd:import
namespace="http://www.wipo.int/standards/XMLSchema/ST96/Common"
schemaLocation="Common_V2_0.xsd"/>
<xsd:import
namespace="http://www.wipo.int/standards/XMLSchema/ST96/Trademark"
schemaLocation="TrademarkApplication_V2_0.xsd"/>
```

4. Define ApplicationCategoryType as indicated below.

```
<xsd:simpleType name="ApplicationCategoryType">
  <xsd:restriction base="xsd:token">
    <xsd:enumeration value="domestic">
      <xsd:annotation>
        <xsd:documentation>Domestic TM
        </xsd:documentation>
      </xsd:annotation>
    </xsd:enumeration>
    <xsd:enumeration value="international">
      </xsd:enumeration>
    </xsd:restriction>
  </xsd:simpleType>
```

5. Declare new element ApplicationCategory and refer to the ApplicationCategoryType simple type defined above.

```
<xsd:element name="ApplicationCategory" type="ustmk:ApplicationCategoryType">
  <xsd:annotation>
    <xsd:documentation>
    Describes type of Trademark application
    </xsd:documentation>
  </xsd:annotation>
</xsd:element>
```

6. Create the new Complex Type TrademarkApplicationType which refers to existing elements except RequestSoftware and add the new element called ApplicationCategory.

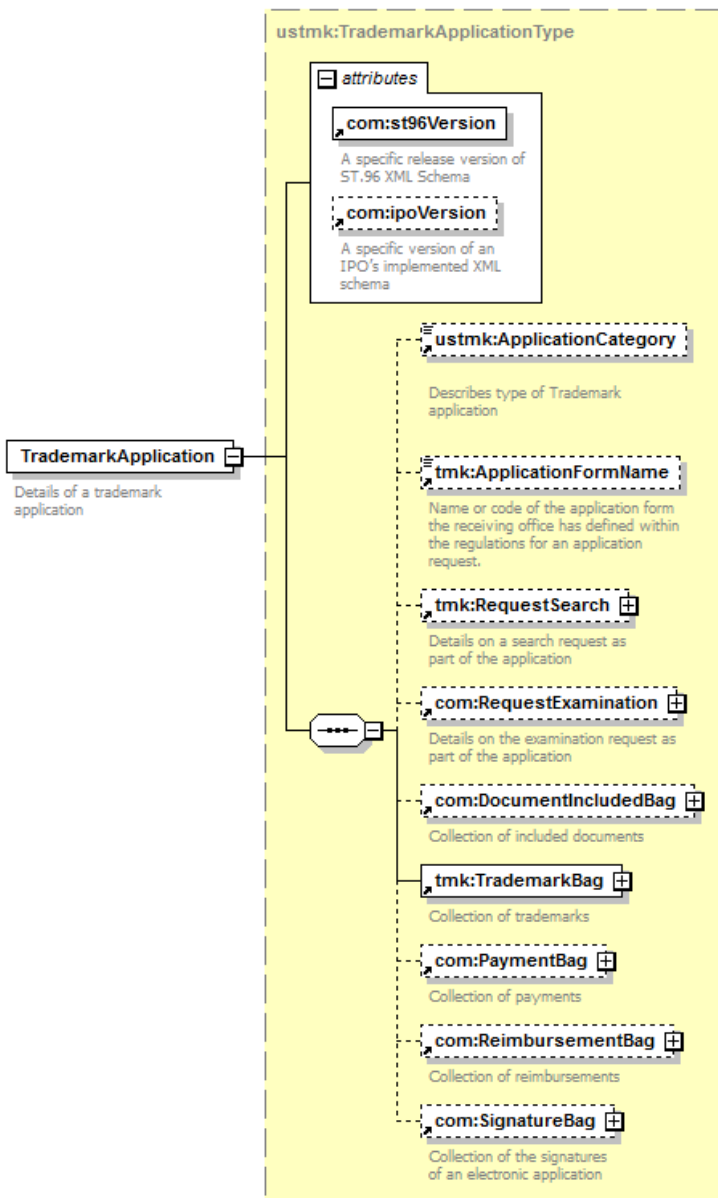
```
<xsd:complexType name="TrademarkApplicationType">
  <xsd:sequence>
    <xsd:element ref="ustmk:ApplicationCategory" minOccurs="0"/>
    <xsd:element ref="tmk:ApplicationFormName" minOccurs="0"/>
    <xsd:element ref="tmk:RequestSearch" minOccurs="0"/>
    <xsd:element ref="com:RequestExamination" minOccurs="0"/>
    <xsd:element ref="com:DocumentIncludedBag" minOccurs="0"/>
    <xsd:element ref="tmk:TrademarkBag"/>
    <xsd:element ref="com:PaymentBag" minOccurs="0"/>
    <xsd:element ref="com:ReimbursementBag" minOccurs="0"/>
  </xsd:sequence>
</xsd:complexType>
```

```
        <xsd:element ref="com:SignatureBag" minOccurs="0"/>
    </xsd:sequence>
    <xsd:attribute ref="com:st96Version" use="required"/>
    <xsd:attribute ref="com:ipoVersion"/>
</xsd:complexType>
```

7. Declare the new element TrademarkApplication which refers to TrademarkApplicationType defined above.

```
    <xsd:element name="TrademarkApplication"
type="ustmk:TrademarkApplicationType">
        <xsd:annotation>
            <xsd:documentation>Details of a trademark
application</xsd:documentation>
        </xsd:annotation>
    </xsd:element>
```

8. The final content model design is shown below:



Generated by XMLSpy

www.altova.com

[Sigue el Anexo VI de la Norma ST.96]

[Sigue el Anexo II]

**ST.96 - ANNEX VI**

TRANSFORMATION RULES AND GUIDELINES

Final Draft

*Proposal presented by the XML4IP Task Force for consideration and adoption at the CWS/4BIS*

**Table of Contents**

1.	INTRODUCTION .....	2
1.1	Overview .....	2
1.2	Scope .....	2
1.3	How to Use this Document .....	2
1.4	Terminology .....	2
1.5	Rule Identifiers .....	2
2.	GUIDANCE FOR DATA PREPARATION .....	2
3.	GUIDANCE FOR MAPPING TABLES .....	3
4.	GUIDANCE FOR DATA CONVERSION .....	4
4.1	Date and Time Format .....	4
4.2	Boolean Values .....	4
4.3	Language and country codes .....	5
4.4	Code and Enumeration Values .....	5
4.4.1	Same Set of Values .....	5
4.4.2	Different Set of Values .....	6
4.5	Identity constraints .....	7
4.6	Different data structure .....	7
4.6.1	Deletion .....	7
4.6.2	Addition .....	7
4.6.3	Renaming .....	7
4.6.4	Change of elements order within sequence .....	7
4.6.5	Empty element .....	8
4.6.6	Change of cardinality .....	8
4.6.7	Change of structure .....	8
4.7	Type Mismatches .....	8
4.7.1	Pattern restriction .....	8
4.7.2	W3C Built-in data types .....	8
	APPENDIX A: ELEMENT AND ATTRIBUTE MAPPING .....	9
	APPENDIX B: ENUMERATION LIST MAPPING .....	9
	APPENDIX C: SAMPLE XSLT CODES .....	10

## 1. INTRODUCTION

### 1.1 Overview

1. Before the adoption of WIPO Standard ST.96, WIPO Standards ST.36, ST.66 and ST.86 had already been used by Industrial Property Offices (IPOs); therefore, maintaining transformability with XML document instances conforming to WIPO Standards ST.36, ST.66 and ST.86 (hereinafter called "existing XML Standards") is one of the primary concerns for WIPO Standard ST.96.

2. For the facilitation of data exchange and interoperability between an IPO using ST.36/ST.66/ST.86 and an IPO using ST.96, bi-directional transformation is desirable. However, it is not realistic to expect perfect bi-directional conversion between instances conforming to ST.96 and instances conforming to ST.36/ST.66/ST.86. Due to improvements based on experience and advances in technology, ST.96 structures will differ in many ways from those defined in ST.36/ST.66/ST.86; therefore, this document aims at defining the necessary degree of transformability between ST.96 and ST.36, ST.66 or ST.86. Moreover, it is noted that bi-directional transformation would be determined on a case-by-case basis.

### 1.2 Scope

3. This document is intended to provide rules and guidelines for transformations between XML instances of ST.96 and instances of WIPO Standards ST.36, ST.66 or ST.86. It does not address either transformations for national implementations or transformation of XML instances of different versions of ST.96.

4. This document includes mapping tables for elements and attributes defined in the Standards in Appendix A of this document and mapping tables for enumerated values and codes specified in the Standards in Appendix B. The mapping tables will be updated in accordance with the evolution of the Standards.

5. This document also provides some examples of eXtensible Stylesheet Language Transformations (XSLT) in Appendix C of this document based on the mapping tables.

### 1.3 How to use this document

6. This document is intended to provide transformation guidance for IPOs that convert their data conforming to ST.36, ST.66 or ST.86 to data conforming ST.96 and *vice versa*.

### 1.4 Terminology

7. In this document:

- the term "*data transformation*" refers to converting data from a source data format into a destination data format. It can be divided into two steps: data mapping and code generation;
- the term "*data mapping*" refers to mapping elements/attributes and codes/enumeration values from the source to the destination; and describing any transformation that must occur. The element/attribute mapping is provided in Appendix A of this document. The code/enumeration list mapping is provided in Appendix B of this document;
- the term "*code generation*" refers to creating transformations in XSLT based on an element mapping specification. Sample XSLT code is provided in Appendix C to this document;
- the term "*input XML instance*" is the XML instance that will be transformed; and
- the term "*output XML instance*" is the XML instance that is the result of transformation.

### 1.5 Rule Identifiers

8. All transformation rules are informative. Transformation rules are identified through a prefix of [TR-*nn*]. The value "*nn*" indicates the sequential number of the rule. For example, the rule identifier [TR-06] identifies the sixth transformation rule (TR).

## 2. GUIDANCE FOR DATA PREPARATION

9. It is expected that an IPO's XML instances are valid against its implementation XML DTD or schema. The implementation XML schema or DTD may define office-specific elements, attributes, types and namespace. In order to utilize the data transformation that converts the XML instances conforming to ST.36, ST.66 or ST.86 to instances conforming to ST.96 and *vice versa*, the Office's XML instances are likely to need to be modified for validation against the corresponding WIPO Standard.



10. In order to prepare valid instances,
- If an IPO uses its own component names instead of using names as defined in the standard, e.g., `WOApplicationBody`, for data preparation, the Office component names in the instances should be renamed to the corresponding component names defined in the Standard; and
  - if no namespace is declared in the instances, the Office should add the namespace declaration in the instances as defined in corresponding Standard.

[TR-01] An Input XML instance should validate against the corresponding XML DTD (`xx-patent-document.dtd` defined in ST.36) or Schema (ST.66, ST.86 and ST.96) of WIPO Standards.

### 3. GUIDANCE FOR MAPPING TABLES

11. The mapping tables are vital to this document. The goal was to specify a one-to-one mapping between each of the elements and attributes of ST.96 and each of the elements and attributes of ST.36, ST.66 and ST.86. This will not be always feasible as explained in Section 1.1. The mapping tables have been created in Appendix A to this document.

12. For each mapping direction (i.e. ST.36 to ST.96, ST.66 to ST.96, ST.86 to ST.96, ST.96 to ST.36, ST.96 to ST.66 and ST.96 to ST.86), a different file is defined. Each file contains two sections, as Appendix A and Appendix B to this document.

13. The following columns exist in each mapping table:

- Input Node
- Output Node
- Type [Cardinality] for Input and Output nodes
- Condition

14. The *Input Node* and *Output Node* column define the atomic element or attribute. For ST.96 components, tag names include namespace prefix. For that reason, the full path for elements or attributes is used. Hierarchical levels represent the path using XPATH notation. The elements are listed in the order of declaration in the corresponding DTD or Schema. For XPath notation, a slash "/" is used to separate the hierarchical levels.

15. The *Type [Cardinality]* column designates:

- the type used by elements or attributes. As ST.36 is defined using DTDs, only the following types are used: ID, CDATA and #PCDATA. For ST.66, ST.86 and ST.96, W3C built-in data-types and user defined data Types are shown in this column. For ST.96 components and other referenced external standards, data Types include the namespace prefix, and;
- the cardinality of elements or attributes:
  - 1..1 = mandatory only one occurrence
  - 0..1 = optional only one occurrence
  - 1..n( $\infty$ ) = mandatory one or many
  - 0..n( $\infty$ ) = optional or many

16. The *Condition* column contains specific mapping or conversion instructions/rules relating to the source and target described in Section "Guidance for Data Conversion".

Input Node	Type [Cardinality]	Condition	Output Node	Type [Cardinality]
ReportCitation/CitedReference/@id	xsd:token [0..1]	TR-23	citation/@id	ID [0..1]

Example of Mapping table

[TR-02] A mapping table SHOULD be developed for each Document level schema defined in ST.96, e.g. `ApplicationBody_V2_0.xsd`, to facilitate data transformation.

[TR-03] Bi-directional transformations SHOULD ensure that the integrity of the data is maintained and that limitations are fully described and supported by IPOs (e.g. in case some data is lost during the conversion).

#### 4. GUIDANCE FOR DATA CONVERSION

17. In addition to the tables for one-to-one mappings, guidance for data conversion is necessary since data formats used by elements or attributes and their content structures in ST.96 may differ from the corresponding elements or attributes in ST.36, ST.66 or ST.86. In ST.96 many elements and attributes originating from ST.36, ST.66 and/or ST.86 have been redefined by simplifying the data structure or using new XML technologies including a data-oriented design approach. The following issues should be considered when converting XML instances and necessary guidance is provided to address the issues below. Details concerning these issues are provided in the subsections that follow:

- Date and time formats;
- Boolean values;
- Language and country codes;
- Code and enumeration values;
- Identity constraints;
- Different data structures; and
- Type mismatches.

##### 1.6 *Date and Time Formats*

18. An ST.36 DTD does not specify a strict data format for dates because of DTD limitations even though the date format of "YYYYMMDD" is recommended. In ST.36, date is defined as #PCDATA even though the format of day/month/year is recommended for some kind of dates, e.g. date of mailing or priority date. In ST.66, ST.86 and ST.96, dates are defined as `xsd:date` which requires the format of YYYY-MM-DD.

19. All of the date fields using a format in which day, month and year information are mandatory can be expressed in a format that is appropriate for ST.96. In such case, the following conversion rules should apply:

[TR-04] To convert to ST.96, the date value should be copied after conversion from the date field of ST.36 into the field of ST.96.

For example, Input in ST.36: date field: 20081025; date format: YYYYMMDD; Output in ST.96: date field 2008-10-25

[TR-05] To convert to ST.36, the date value should be copied from the field of ST.96 into the date field of ST.36 but without the dashes.

For example, Input in ST.96 date value: 2008-10-25, and Output in ST.36: date field: 20081025.

20. ST.36 does not specify a strict format for time fields because of DTD limitations. In ST.36, time is defined as #PCDATA even though the format of HHMM is recommended. In ST.66, ST.86 and ST.96, time fields are defined as `xsd:time` which requires the format of hh:mm:ss.

21. All of the time fields using a format in which hour, minute and second information is mandatory can be expressed in a format that is appropriate for ST.96. In such cases, the following conversion rules should apply:

[TR-06] To convert to ST.96, the ST.36 time value should be copied, after conversion, into the time value of ST.96.

For example, Input in ST.36: time value 1030; time format: HHMM; Output in ST.96: time value 10:30:00

[TR-07] To convert to ST.36, the ST.96 time value should be copied into the time value of ST.36 but without colons or seconds.

For example, Input in ST.96 time value: 10:30:00, and Output in ST.36: time value: 1030.

##### 1.7 *Boolean Values*

22. ST.36 does not specify a strict format for Boolean values because of DTD limitations. It is assumed that each IPO has a specific business practice to consistently define the Boolean format in its XML instances. The following values, '0' / '1', 'no' / 'yes', or 'false' / 'true', are commonly used as Boolean values in ST.36 XML instances. In ST.96, 'false' / 'true' Boolean values are used. For the conversion of Boolean values between ST.36 and ST.96 instances, the following rules should apply:

- [TR-08] To convert to ST.36, a Boolean value should be copied from the ST.96 field into the Boolean field of ST.36. If necessary, IPOs should adapt the values according to their practices.
- [TR-09] To convert to ST.96, a Boolean value from the ST.36 field should be converted to either 'true' or 'false' and copied into the Boolean field of ST.96.

23. In order to facilitate the conversion of Boolean values, the sample XSLT code in Appendix C includes a conversion scheme so that the following common values: '0' / '1', 'no' / 'yes', or 'false' / 'true' in an ST.36 XML instance are converted to 'true' or 'false' in ST.96. If an ST.36 instance contains the other values, IPO should convert the values to one of common values and run the XSLT script.

#### 1.8 Language and country codes

24. In ST.36, the language element/attribute is defined as alpha-numeric (#PCDATA), even though ISO two-letter codes are recommended in the description of the corresponding element/attribute. Therefore, if ISO two-letter language codes are not used in ST.36 XML instances, the data of language code in the instances should be converted into corresponding two-letter code before transformation.

25. In ST.36, the country element/attribute is defined as alpha-numeric (#PCDATA), even though ISO two-letter country codes are recommended in the description of corresponding element/attribute. While both standards use ISO Country codes, in ST.36, lower case country codes are allowed, while in ST.96 upper case codes are mandatory; therefore the codes' case should be changed to match that required by each standard during transformation.

- [TR-10] To convert to ST.96, the ST.36 field value for country or language should be case-transformed to the corresponding values defined in ST.96, and the result code should be inserted into the ST.96 field.

- [TR-11] To convert to ST.36, the ST.96 code value for country or language should be inserted into the ST.36 field.

#### 1.9 Code and Enumeration Values

26. ST.36, ST.66, ST.86 and ST.96 provide sets of codes and enumerated values. In many cases, ST.96 has the same code or enumerated values as defined in ST.36, ST.66 or ST.86. In other cases, however, ST.96 defines additional or different codes/ enumerated values from the other Standards, in order to reflect IPOs' practice and/or follow the ST.96 DRCs. Guidance is provided on the conversion of the codes, depending on whether ST.96 supports the same codes or whether additional code values have been introduced.

27. Some ST.36 fields are defined as #PCDATA rather than as a list of specific enumeration values whereas in ST.96, an enumerated list or code is used in most cases. In ST.66/ST.86, some types are defined as a union of a free format and an enumerated list.

- [TR-12] To convert to ST.96, the ST.66/ST.86 field value should be mapped to the list of ST.96 codes, and the mapped code should be inserted into the ST.96 field. The free format value should be ignored unless it is possible to map it to an ST.96 code.

- [TR-13] To convert to ST.96, the ST.36 field value should be ignored unless it is possible to map it to an ST.96 code. If the value does not conform to the expected data type in ST.96, it may have to be reformatted so that the instance will validate successfully.

- [TR-14] To convert to ST.36/ST66/ST86, the ST.96 code value should be mapped to the field of corresponding Standard.

**WARNING:** When copying values from ST.36, ST.66 or ST.86 to ST.96 (or the reverse), or when revising ST.36, ST.66 or ST.86 to incorporate values from ST.96, consider the possibility that a newly added value might violate a business rule in the new context.

#### Same Set of Values

28. In most cases, ST.36/66/86 and ST.96 expect the same set of codes or enumerated values for some specific elements or attributes listed in Appendix B of this document.

For example:

ST.66 component	ST.66 allowed values	ST.96 component	ST.96 allowed values
MarkKind	Individual	MarkKind	Individual
	Collective		Collective
	Certificate		Certificate
	Guarantee		Guarantee
	Defensive		Defensive
	Other		Other

29. In this case, the following recommendation applies:

[TR-15] To convert to ST.96, the field value should be copied from the ST.36/66/86 field into the ST.96 field. The formatting of the value should follow the conventions specified in the ST.96 DRCs.

[TR-16] To convert to ST.36/66/86, the field value should be copied from the ST.96 field into the ST.36/66/86 field. The formatting of the value should follow the conventions specified in ST.36, ST.66 or ST.86.

30. In the case that the elements or attributes are required in ST.96 and optional in ST.36, ST.66 and ST.86, the following conversion rules below apply:

[TR-17] If the ST.36/ST.66/ST.86 component is populated, the field value should be copied from the ST.36/ST.66/ST.86 field into the ST.96 field. The formatting of the value should follow the conventions specified in the ST.96 DRCs.

[TR-18] If the ST.36/ST.66/ST.86 component is not populated, the ST.96 component should be populated with the "Undefined" value.

31. There is no case where the element or attribute required in ST.36, ST.66 and ST.86 is defined as optional in ST.96; no rule, therefore, is specified in this document for this circumstance.

#### Different Set of Values

32. In some cases listed in Appendix B of this document, ST.96 defines different codes or enumerated values from ST.36, ST.66 or ST.86, but the codes or values have the same meaning. For example:

ST.36 component	ST.36 allowed values	ST.96 component	ST.96 allowed values
Orient	Port	orientation	Portrait
	Land		Landscape

33. In some cases, ST.96 defines more codes or values than in ST.36/ST.66/ST.86. In these cases, transformation from ST.36/ST.66/ST.86 to ST.96 is not an issue. However, transformation from ST.96 to ST.36/ST.66/ST.86 is not ensured. In order to guarantee the transformation from ST.96 to ST.36/ST.66/ST.86, the existing XML Standards should be revised to capture the additional codes or values defined in ST.96. For example:

ST.36 component	ST.36 allowed values	ST.96 component	ST.96 allowed values
Color	color	ColourMode	Colour
	Bw		Black and white
	-		Greyscale

34. In general, an ST.96 component does not define fewer values than the corresponding components defined in ST.36, ST.66 or ST.86. For this reason, rules regarding this case are not provided in this document.

- [TR-19] To convert to ST.96, a value should be copied from the ST.36, ST.66 or ST.86 field into the ST.96 field, if the both values have the same meaning.
- [TR-20] To convert to ST.36/ST.66/ST.86, a value should be copied from the ST.96 field into the ST.36/ST.66/ST.86 field, if the both values have the same meaning.
- [TR-21] If the corresponding value does not exist in destination standard, the value should be copied unchanged in to the output instance even though the value is invalid against the destination standard schema or DTD.

#### 1.10 Identity constraints

35. ST.96 recommends using `xsd:key/xsd:unique/xsd:keyref` and/or `xsd:ID/xsd:IDREF/xsd:IDREFS` for identity constraints while ST.36 uses ID/IDREF/IDREFS. Some attributes in ST.36 employ the ID and IDREF/IDREFS types: For example, `citation/id` uses ID type. The values starting with a numeric character are not allowed in `xsd:ID/xsd:IDREF/xsd:IDREFS` field.

- [TR-22] To convert to ST.96, the value of the ST.36/ST.66/ST.86 field associated with `xsd:ID/xsd:IDREF/xsd:IDREFS` type should be copied to the corresponding ST.96 field.
- [TR-23] To convert to ST.36/ST.66/ST.86, the `xsd:ID/xsd:IDREF/xsd:IDREFS` value of the ST.96 field should be copied to the corresponding ST.36/ST.66/ST.86 field. The `xsd:key/xsd:unique/xsd:keyref` value of the ST.96 field should be copied to the ST.36/ST.66/ST.86 field ensuring no duplication of ID values in the instance and revising any corresponding IDREF and IDREFS. The `xsd:key/xsd:unique/xsd:keyref` values starting with a numeric character should be converted to non-numeric character values..

#### 1.11 Different data structures

36. ST.96 provides different structures for some components defined in ST.36, ST.66 or ST.86. Guidance is provided on conversion from one structure to the other. Three major changes can occur in this context: deletion, addition, and renaming.

##### Deletion

37. Some elements or attributes defined in ST.36, ST.66 or ST.86 have been deleted in ST.96 as they are no longer used. They are marked as "NOT USED" in Appendix A of this document. For example, the following elements `citation`, `nplicit`, `article`, `book` and `text` are no longer used and therefore have no corresponding element in ST.96.

- [TR-24] To convert to ST.96, the ST.36/ST.66/ST.86 field should be ignored as there is no counterpart component in ST.96.
- [TR-25] To convert to ST.36/ST.66/ST.86 the field should not be provided in ST.36/ST.66/ST.86 as the related field is optional in ST.36/ST.66/ST.86.

##### Addition

38. Some elements or attributes have newly been added in ST.96 which have no counterpart in ST.36, ST.66 or ST.86.

- [TR-26] All IPO specific elements and attributes should be ignored in the transformation process.
- [TR-27] In the case of the addition of an optional element or attribute, to convert to ST.96, the ST.96 field should not be populated, as there is no counterpart component in ST.36/ST.66/ST.86.
- [TR-28] When adding a mandatory element or attribute, a mandatory element should not be created if the corresponding optional element or attribute is not present in the input instance. An error message should be given in the validation process of the output instance. The IPOs can decide how to resolve the differences on a case-by-case basis.

##### Renaming

39. Almost all fields have been renamed in ST.96. A one-to-one mapping is provided in Appendix A of this document. For example, the `absno` element in ST.36 is mapped to ST.96 `AbstractNumber`.

- [TR-29] To convert to ST.96, the value of the ST.36/ST.66/ST.86 field should be copied to the ST.96 field.
- [TR-30] To convert to ST.36/ST.66/ST.86, the value of the ST.96 field should be copied to the ST.36/ST.66/ST.86 field.

##### Change of elements order within a sequence construct

40. In ST.96 Schema, some elements have a different order for child elements from the corresponding elements defined in ST.36, ST.66 or ST.86. In a sequence construct, the order of child elements is important therefore the changed order of child elements within the sequence should be considered when transformation is performed.

### Empty element

41. Since an empty element is not allowed in ST.96, a corresponding one-to-one mapping element to empty element within ST.36, ST.66 and ST.86 cannot be defined. Empty elements have been defined in various ways in the individual standards; therefore the transformation of these elements will differ on a case by case basis.

42. In ST.36, in some cases, elements indicating the existence of information are empty elements and the corresponding elements in ST.96 are defined as `xsd:boolean` Type. In other cases, the presence of some empty elements in an ST.36 XML instance can be mapped to an enumeration value in the ST.96 XML instance.

43. In the ST.66 model schema and ST.86 model schema, most elements have no mandatory child element, which means the parent elements can have empty content. In order to avoid empty content the `sequence` construct in the elements is changed to a multiple `choice` construct in corresponding elements defined in ST.96. This kind of structural change should be considered when the transformation is designed.

### Change of cardinality

44. An element may be defined as optional in one Standard and as mandatory in another.

[TR-31] When changing the cardinality of an element from optional to mandatory, a mandatory element should not be created if the corresponding optional element is not present in the input instance. An error message should be given in the validation process of the output instance. The IPOs can decide how to resolve the differences.

### Change of structure

45. For clarity, ST.96 proposes a different structure from the one defined in ST.36/ST.66/ST.86. For these cases, some conditions should be set for transformation on a case-by-case basis. For example,

- if `doc-page` is used in the context of drawings, check the type attribute.
  - If `@type=jpg` or `tif`, map with `PageImage`.
  - If `@type=pdf`, map with `DocumentURI`.
- in other contexts, check the presence of the `ocr` attribute.
  - If the `ocr` is defined, map with `DocumentURI`,
  - Otherwise map with `PageImage`.

### 1.12 *Type Mismatches*

46. In ST.96, some elements or attributes have type restrictions, but the ST.36 DTD has very few typed attributes or elements. Only `#PCDATA`, `ID`, `IDREF` and `CDATA` for atomic elements or attributes are used.

### Pattern restriction

47. Some ST.36 fields are defined as `#PCDATA` and therefore do not contain pattern restrictions. These fields can easily accept the more restricted values of ST.96.

[TR-32] To convert to ST.96, the ST.36 field should be copied as-is to the ST.96 field. If the value does not conform to a pattern restriction in ST.96, it may have to be reformatted so that the instance will validate successfully.

[TR-33] To convert to ST.36, the ST.96 field value should be copied as-is into the free-text field of ST.36.

### W3C Built-in data types

48. In principle, un-typed atomic elements or attributes defined in a DTD are mapped to `xsd:string` where a type is required. ST.96 uses the following W3C Built-in data types: `xsd:token`, `xsd:positiveInteger`, `xsd:boolean` and `xsd:string`. Although the format defined for some fields in ST.36 is alpha-numeric (`#PCDATA`), their values are expected to follow the W3C built-in data types in ST.96.

[TR-34] To convert to ST.96, the ST.36 field should be copied as-is to the ST.96 field. If the value does not conform to the expected data type in ST.96, it may have to be reformatted so that the instance will validate successfully.

[TR-35] To convert to ST.36, the ST.96 field value should be copied as-is into the free-text field of ST.36.

## APPENDICES

The Appendixes are available at: <http://www.wipo.int/standards/en/st96/v2-0/annex-vi>

### APPENDIX A: ELEMENT AND ATTRIBUTE MAPPING

Appendix A aims at providing a model one-to-one mapping between ST.96 elements and attributes and the corresponding elements and attributes of ST.36, ST.66 and ST.86. The one-to-one mapping is not always achieved due to reasons explained in Annex VI to ST.96, Transformation Rules and Guidelines; therefore, Appendix A is intended to provide a mapping between ST.96 and ST.36, ST.66 or ST.86 to the necessary degree.

The following mapping tables are provided in the corresponding file:

- Mapping Table for Elements and Attributes regarding the transformation of ST.36 application-body to ST.96 `ApplicationBody`
- Mapping Table for Elements and Attributes regarding the transformation of ST.96 `ApplicationBody` to ST.36 application-body
- Mapping Table for Elements and Attributes regarding the transformation of ST.36 bibliographic-data to ST.96 `BibliographicData`
- Mapping Table for Elements and Attributes regarding the transformation of ST.96 `BibliographicData` to ST.36 bibliographic-data
- Mapping Table for Elements and Attributes regarding the transformation of ST.66 `TradeMark` to ST.96 `Trademark`
- Mapping Table for Elements and Attributes regarding the transformation of ST.96 `Trademark` to ST.66 `TradeMark`
- Mapping Table for Elements and Attributes regarding the transformation of ST.86 Design to ST.96 Design
- Mapping Table for Elements and Attributes regarding the transformation of ST.96 Design to ST.86 Design

### APPENDIX B: ENUMERATION LIST MAPPING

Appendix B aims at providing a model one-to-one mapping for codes or enumerated values between WIPO Standard ST.96, and WIPO Standards ST.36, ST.66 or ST.86. The one-to-one mapping is not always achieved due to reasons explained in Annex VI to ST.96, Transformation Rules and Guidelines.

The following mapping tables are provided in the corresponding file:

- Mapping Table for Enumeration List regarding the transformation of ST.36 application-body to ST.96 `ApplicationBody`
- Mapping Table for Enumeration List regarding the transformation of ST.96 `ApplicationBody` to ST.36 application-body
- Mapping Table for Enumeration List regarding the transformation of ST.36 bibliographic-data to ST.96 `BibliographicData`
- Mapping Table for Enumeration List regarding the transformation of ST.96 `BibliographicData` to ST.36 bibliographic-data
- Mapping Table for Enumeration List regarding the transformation of ST.66 `TradeMark` to ST.96 `Trademark`
- Mapping Table for Enumeration List regarding the transformation of ST.96 `Trademark` to ST.66 `TradeMark`
- Mapping Table for Enumeration List regarding the transformation of ST.86 Design to ST.96 Design
- Mapping Table for Enumeration List regarding the transformation of ST.96 Design to ST.86 Design

#### APPENDIX C: SAMPLE XSLT CODES

Appendix C aims at providing sample XSLT (Extensible Stylesheet Language Transformations) codes for data conversion between an ST.96 instance and an ST.36, ST.66 or ST.86 instance based on Appendixes A and B.

- Sample XSLT Codes for “ApplicationBody”, “BibliographicData”, “Trademark” and “Design”: The conversion stylesheets consist of a set of files that can be used to convert ST.96 instances to ST.36, ST.66, or ST.86 instances and vice versa.

[Fin del Anexo VI y de la Norma]

[Fin del Anexo II y del documento]