Draft XML4IP Design Rules comparing WIPO Standard ST.86

Draft Proposal for XML4IP Design Rules	WIPO Standard ST.86
REQUIREMENTS OF THE STANDARD	**REQUIREMENTS OF THE STANDARD**
15. Schemas are developed for document types based on the global ICEs, IP-type ICEs and office-specific extension (defined in “Dictionary”: proper terminology to be found)	15. ST.86 XML Dictionary in Appendix A is the foundation of this Standard.
16. The Dictionary MUST be used as defined in this Standard, that is, the types, elements, attributes and enumerations MUST be as indicated in the list of the Dictionary.	16. The Dictionary MUST be used as defined in this Standard, that is, the types, elements, attributes and enumerations MUST be as indicated in the list of the Dictionary. However, some enumerations are defined as open and may be restricted or extended in the specific offices’ implementation.
17. Implementation that conforms to this Standard MUST be carried out according to the guidelines in this Standard.	17. Implementation that conforms to this Standard MUST be carried out according to the guidelines in this Standard, or MUST be an extension of a conforming XSD according to the guidelines in this Standard.
18. Document instances that conform to this Standard MUST be well-formed XML.. In addition, such instances MUST be valid against schemas of document types developed by office-specific extension methods as explained in paragraph 15 (above). XML4IP component schemas which serve as the base of office-specific extension methods are listed in Appendix B.	18. XML instances that conform to this Standard MUST be well-formed XML and validated by the XSD in Appendix B.
19. It is understood that this Standard cannot possibly include all elements required by all offices; in such implementation schema, office-specific elements are allowed as described below in the “Naming Office-Specific Types and Elements” section.	19. It is understood that this Standard cannot possibly include all elements required by all industrial design offices; in such implementation schema, office-specific elements are allowed as described below in the “Naming Office-Specific Types and Elements” section.

20. The W3C XML Schema definition language has become the generally accepted schema language that is experiencing the most widespread adoption. Although other schema languages exist that offer their own advantages and disadvantages, all XML Schema design rules MUST be based on the W3C XML Schema Recommendations: XML Schema Part 1: Structures and XML Schema, Part 2: Datatypes. All schemas and messages MUST be based on the W3C suite of technical specifications holding recommendation status.	20. The W3C XML Schema definition language has become the generally accepted schema language that is experiencing the most widespread adoption. Although other schema languages exist that offer their own advantages and disadvantages, all XML Schema design rules MUST be based on the W3C XML Schema Recommendations: XML Schema Part 1: Structures and XML Schema, Part 2: Datatypes. All schemas and messages MUST be based on the W3C suite of technical specifications holding recommendation status.
21. Redefinition of XSD built-in data types SHOULD be avoided.	21. Redefinition of XSD built-in data types SHOULD be avoided.
22. WIPO Standard ST.3 MUST be used for priority country, seniority country, designated country, contracting party and receiving Office	22. WIPO Standard ST.3 MUST be used for priority country, contracting party, receiving Office and designated Office.
23. ISO 3166 MUST be used for address country codes, exhibition country codes and nationalities.	23. ISO 3166 MUST be used for address country codes, exhibition country codes and nationalities.
24. ISO/IEC 10646 . UCS . Unicode UTF-8 MUST be used for character set.	24. ISO/IEC 10646 . UCS . Unicode UTF-8 MUST be used for character set.
25. ISO 639-1 (2-Letter Language Codes) MUST be used for Language Codes.	25. ISO 639-1 (2-Letter Language Codes) MUST be used for Language Codes.
26. ISO 8601 - International Standard Date and Time Notation MUST be used for Date and Time Notation. W3C Schema data types include date and time and SHOULD be used in preference to ISO 8601, where there is any conflict.	26. ISO 8601 - International Standard Date and Time Notation MUST be used for Date and Time Notation. W3C Schema data types include date and time and SHOULD be used in preference to ISO 8601, where there is any conflict.
27. ISO 4217-Alpha (3-Letter Currency Codes) MUST be used for Currency Codes.	27. ISO 4217-Alpha (3-Letter Currency Codes) MUST be used for Currency Codes.
Characters	**Characters**

28. This Standard recommends Unicode exclusively. It may be useful to add character entities for characters not yet in Unicode, Use of these entities requires the creation of glyphs for presentation, which do not yet exist. See http://www.w3.org/XML/Core/2002/10/charents-20021023 for further information about character entities.	28. This Standard recommends Unicode exclusively. It may be useful to add character entities for characters not yet in Unicode, Use of these entities requires the creation of glyphs for presentation, which do not yet exist. See http://www.w3.org/XML/Core/2002/10/charents-20021023 for further information about character entities.
29. Document instances MUST include an XML declaration as the first line in the file. <?xml version='1.0' encoding='utf-8' ?>	29. Document instances MUST include an XML declaration as the first line in the file. <?xml version='1.0' encoding='utf-8' ?>
Note that only UTF-8 is recommended in this Standard. However, in the case of ideographic scripts, Unicode in UTF-8 MAY produce exceptionally large files since the encoding MAY use up to four bytes per character. In such cases, national Offices MAY select an encoding that brings files to manageable sizes. Offices that elect to do so, SHOULD be prepared to consult with their exchange partners and to give adequate public notice.	Note that only UTF-8 is recommended in this Standard. However, in the case of ideographic scripts, Unicode in UTF-8 MAY produce exceptionally large files since the encoding MAY use up to four' bytes per character. In such cases, national Offices MAY select an encoding that brings files to manageable sizes. Offices that elect to do so, SHOULD be prepared to consult with their exchange partners and to give adequate public notice.
30. The characters that are permitted to appear in an XML document are specified in the XML 1.0 W3C Recommendation, and are endorsed by this Standard with the following exception. The characters used in type, element, or attribute names described in this Standard are restricted to the following set: {a-z, A-Z, 0-9, period (.), dash (-) and underscore (_)}.	30. The characters that are permitted to appear in an XML document are specified in the XML 1.0 W3C Recommendation, and are endorsed by this Standard with the following exception. The characters used in type, element, or attribute names described in this Standard are restricted to the following set: {a-z, A-Z, 0-9, period (.), dash (-) and underscore (_)}.
General XML Constructs	**General XML Constructs**
Naming and Modeling Constraints	**Naming and Modeling Constraints**

Naming Constraints	**Naming Constraints**
31. Each dictionary entry name MUST define one and only one fully qualified path for an element or attribute.	31. Each dictionary entry name MUST define one and only one fully qualified path for an element or attribute.
Modeling Constraints	**Modeling Constraints**
32. Libraries and Schemas MUST only use approved datatypes. This (rule) does not apply to country-specific extension components.	32. Libraries and Schemas MUST only use approved datatypes.
33. deleted	33. Mixed content MUST NOT be used in data centric schema except where contained in an xsd:documentation element.
Reusability Scheme	**Reusability Scheme**
34. All type declaration MUST be global.	34. All type declaration MUST be global.
Namespace Scheme	**Namespace Scheme**
Declaring Namespaces	**Declaring Namespaces**
35. Every schema module, except internal schema modules, MUST have a namespace declared using the xsd:targetNamespace attribute.	35. Every schema module, except internal schema modules, MUST have a namespace declared using the xsd:targetNamespace attribute.
36. All XML schemas MUST declare the W3C schema namespace. Schemas MUST declare a target namespace.	36. All XML schemas MUST declare the W3C schema namespace. Schemas MUST declare a target namespace.
37. Namespace qualification MUST be used for W3C schema construct.	37. Namespace qualification MUST be used for W3C schema construct.
38. Namespaces MUST not be changed even if versions are changed. Instead, version information MUST be described in the root element of instances and schemas. Further, names of schema files MUST include some sort of information that indicates versions. Follow guidelines in "Appendix **: Versioning Management of Schemas". (to be researched)	38. Every defined or used schema set version MUST have its own unique namespace.

39. Published namespaces MUST never be changed.	39. Published namespaces MUST never be changed.
40. There SHALL be no default namespaces used in Schema definitions. (to be researched whether instances have namespace as well) That is, for example, both the XMLSchema and targetNamespace MUST be explicitly qualified. This approach, even though quite cluttered, is more consistent for all types of schema with no, one, or multiple targetNamespaces. For example:	40. There SHALL be no default namespaces. That is, for example, both the XMLSchema and targetNamespace MUST be explicitly qualified. This approach, even though quite cluttered, is more consistent for all types of schema with no, one, or multiple targetNamespaces. For example:
<?xml version=”1.0”?>	<?xml version=”1.0”?>
<xsd:schema xmlns:xsd=”http://www.w3.org/2001/XMLSchema”	<xsd:schema xmlns:xsd=”http://www.w3.org/2001/XMLSchema”
targetNamespace=”http://www.wipo.int/standards/XMLSchema”	targetNamespace=”http://www.wipo.int/standards/XMLSchema”
xmlns:lib=”http://www.wipo.int/standards/XMLSchema”	xmlns:lib=”http://www.wipo.int/standards/XMLSchema”
elementFormDefault=”qualified”>	elementFormDefault=”qualified”>
<xsd:include schemalocation=”xxx.xsd”/>	<xsd:include schemalocation=”xxx.xsd”/>
</xsd:schema>	</xsd:schema>
41. To hide or expose Namespaces in instance documents, the binary switch attribute SHOULD be used: elementFormDefault of the element <xsd:schema> (qualified or unqualified).	41. To hide or expose Namespaces in instance documents, the binary switch attribute SHOULD be used: elementFormDefault of the element <xsd:schema> (qualified or unqualified).
42. External schema references SHOULD use the “Import” construct.--	42. External schema references SHOULD use the “Include” construct. The including and included schemas MUST have the same target namespace.
43. Separate namespaces SHOULD be specified for distinction for XML4IP Core Schemas, Reference Schemas and country-specific schemas. Number of nameSpaces should be kept to a minimum.	43. For simplicity, a single namespace configuration SHOULD be preferred. Multiple namespaces MAY be used for extension purposes (nationalization).

Naming Conventions	**Naming Conventions**
XML Tag Naming Rules	**XML Tag Naming Rules**
44. The XML tag naming conventions are based on the concepts as defined in ISO 11179 Part 5. Element, attribute and type names SHOULD consist of the Object Class, the name of the Property Term and the name of a Representation Term.	44. The XML tag naming conventions are based on the concepts as defined in ISO 11179 Part 5. Element, attribute and type names SHOULD consist of the Object Class, the name of the Property Term and the name of a Representation Term.
(a) An Object Class identifies the primary concept of the element. It refers to an activity or an object within a business context and MAY consist of one, two or three words.	(a) An Object Class identifies the primary concept of the element. It refers to an activity or an object within a business context and MAY consist of one, two or three words.
(b) The Property Term identifies the characteristics of the object class. The name of a Property Term SHALL occur naturally in the tag definition and MAY consist of one, two or three words. A name of a Property Term SHALL be unique within the context of an Object Class but MAY be reused across different Object Classes.	(b) The Property Term identifies the characteristics of the object class. The name of a Property Term SHALL occur naturally in the tag definition and MAY consist of one, two or three words. A name of a Property Term SHALL be unique within the context of an Object Class but MAY be reused across different Object Classes.
(c) If the Representation Term uses the same word as the last one used by the Property Term, the Representation Term SHALL be omitted.	(c) If the Representation Term uses the same word as the last one used by the Property Term, the Representation Term SHALL be omitted.
(d) Terms used as properties names MAY be omitted when the Property Term alone is commonly used and sufficient to express the concept without confusion in its context.	(d) Object class and Representation Terms SHOULD be omitted when the Property Term alone is commonly used and sufficient to express the concept without confusion in its context.
	(e) For example (Object Class + Property Term + Representation Term):
	-ApplicantNationalityCode: Applicant(Object Class) + Nationality(Property Term) + Code(Representation Term)
	-ViewFilename: View(Object Class) + Filename(Property Term) + Text(Representation Term, omitted)

	-FilingDate: Mark(Object Class, omitted) + FilingDate(Property Term)+ Date(Representation Term, omitted)
45. Element, attribute and type names MUST be unique. The names SHOULD be concise and SHOULD NOT contain consecutive redundant words, and MUST be as much as possible self-described and highly structured.	45. Element, attribute and type names MUST be unique. The names SHOULD be concise and SHOULD NOT contain consecutive redundant words, and MUST be as much as possible self-described and highly structured.
46. Element, attribute and type names and all their components MUST be in singular form unless the concept itself is plural.	46. Element, attribute and type names and all their components MUST be in singular form unless the concept itself is plural.
47. Element, attribute and type names MUST only contain nouns, adjectives and eventually verbs. Words like “and”, “of”, “the” MUST be removed, except when this makes the name misleading.	47. Element, attribute and type names MUST only contain nouns, adjectives and eventually verbs. Words like “and”, “of”, “the” MUST be removed, except when this makes the name misleading. For example, IndicationProduct, OpenToLicencingIndicator.
48. Element, attribute and type names MUST NOT be translated, changed or replaced for any purpose.	48. Element, attribute and type names MUST NOT be translated, changed or replaced for any purpose.
49. Element, attribute and type names MUST be composed of words in the English language, using the primary English spellings provided in the Oxford English Dictionary, including office-specific tags (except see paragraph 57 below for acronyms).	49. Element, attribute and type names MUST be composed of words in the English language, using the primary English spellings provided in the Oxford English Dictionary, including office-specific tags (except see paragraph 57 below for acronyms).
50. Element names MUST be in upper camel case (UCC). UCC style capitalizes the first character of each word which compounds the name. For example, AddressCountryCode.	50. Element names MUST be in upper camel case (UCC). UCC style capitalizes the first character of each word which compounds the name. For example, AddressCountryCode.
51. Type names MUST be in UCC + Suffix Type. For example, LanguageCodeType.	51. Type names MUST be in UCC + Suffix Type. For example, LanguageCodeType.
52. Attribute names MUST be in lower camel case (LCC). LCC style capitalizes the first character of each word except the first word. For example, currencyCode=”EUR”.	52. Attribute names MUST be in lower camel case (LCC). LCC style capitalizes the first character of each word except the first word. For example, currencyCode=”EUR”.

53. Regarding the enumeration of values or code list text, it SHOULD be short but semantically sufficient and in English when there is no standard code list. The values and codes SHOULD be drawn from the common industrial property business language.	53. Regarding the enumeration of values or code list text, it SHOULD be short but semantically sufficient and in English when there is no standard code list. The values and codes SHOULD be drawn from the common industrial property business language.
54. A limit of 35 characters for a name is recommended. When the same word is repeated in an element name, the second or following occurrences SHOULD be removed.	54. A limit of 35 characters for a name is recommended. When the same word is repeated in an element name, the second or following occurrences SHOULD be removed.
55. Element, attribute and type names MUST NOT include periods (.), spaces or other separators, or characters not allowed by W3C XML 1.0 for XML names except as specified in this Standard. For example, Office or domain prefixes (XX_UCC with XX in ST.3 code).	55. Element, attribute and type names MUST NOT include periods (.), spaces or other separators, or characters not allowed by W3C XML 1.0 for XML names except as specified in this Standard. For example, Office or domain prefixes (XX_UCC with XX in ST.3 code).
56. The characters used in enumeration value names described in this Standard are restricted to the following set: {a-z, A-Z, 0-9, period (.), comma (,), spaces, dash (-) and underscore (_)}.	56. The characters used in enumeration value names described in this Standard are restricted to the following set: {a-z, A-Z, 0-9, period (.), comma (,), spaces, dash (-) and underscore (_)}.
Acronyms and Abbreviations	**Acronyms and Abbreviations**
57. XML element, attribute and type names MUST NOT use acronyms, abbreviations, or other word truncations, except as specified in this Standard or listed in Appendix D. 1. CV2: Card Verification Value 2. ISO: International Standard Organization 3. PKCS7: Public Key Cryptography Standard 7 4. ST3: WIPO Standard ST.3 5. URI: Uniform Resource Identifier 6. URL: Uniform Resource Locator 7. XSD: XML Schema Definition 8. BW Black and White 9. IPR Intellectual Property Right * This is listed in ST.86 Appendix D. This includes the list in ST.66	57. XML element, attribute and type names MUST NOT use acronyms, abbreviations, or other word truncations, except as specified in this Standard or listed in Appendix D.

	58. The acronyms and abbreviations listed in Appendix D MUST always be used instead of the complete extended name.
59. Acronyms and abbreviations at the beginning of an attribute declaration MUST appear in all lower case. All other acronym and abbreviation usage in an attribute declaration MUST appear in upper case.	59. Acronyms and abbreviations at the beginning of an attribute declaration MUST appear in all lower case. All other acronym and abbreviation usage in an attribute declaration MUST appear in upper case.
60. Acronyms MUST appear in all upper case for all element declarations and type definitions.	60. Acronyms MUST appear in all upper case for all element declarations and type definitions.
XML Schema File naming Rules	**XML Schema File naming Rules**
	61. These conventions will ensure that objects will be stored in a manner that will ensure consistency, uniformity, and comprehensiveness, and will be suitable for all aspects of storage and reuse.
62. Schema and style sheet filenames are recommended to follow a six-part naming rule. The six-part naming rule is illustrated below:	62. Schema and style sheet filenames are recommended to follow a six-part naming rule. The six-part naming rule is illustrated below:
63. File names SHOULD follow the above-mentioned tag naming rules. However a mapping can be defined locally if the rules cannot be applied due to technical constraints. Such local rules MUST be well defined and published for all potential users.	63. File names SHOULD follow the above-mentioned tag naming rules. However a mapping can be defined locally if the rules cannot be applied due to technical constraints. Such local rules MUST be well defined and published for all potential users.

64. Schema file names SHOULD have their versions changed when an included modular schema is updated. With regard to the latest version of the XML schema, offices MAY provide two types of the latest version on their web site: a versioned schema which has an appropriated version number (e.g., WIPOST3Code-V2005-05-21.xsd, st36for66-V1-0.xsd) and a non-versioned schema without version number (e.g., WIPOST3Code.xsd, st36for66.xsd), which is a copy of (or refers to) the latest version of the XML schema. Other offices can refer to the latest version by indicating either the versioned schema with the latest version number or the non-version schema which is a copy of the latest version to avoid changing office's implemented codes whenever the schema is updated.	64. Schema file names SHOULD have their versions changed when an included modular schema is updated. With regard to the latest version of the XML schema, offices MAY provide two types of the latest version on their web site: a versioned schema which has an appropriated version number (e.g., WIPOST3Code-V2005-05-21.xsd, st36for66-V1-0.xsd) and a non-versioned schema without version number (e.g., WIPOST3Code.xsd, st36for66.xsd), which is a copy of (or refers to) the latest version of the XML schema. Other offices can refer to the latest version by indicating either the versioned schema with the latest version number or the non-version schema which is a copy of the latest version to avoid changing office's implemented codes whenever the schema is updated.
Miscellaneous XSD Rules	**Miscellaneous XSD Rules**
65. Restriction on field length MUST not be defined in ST XML Schema set by WIPO. However, it MAY be defined for other than basic components.	65. Restriction on field length MUST not be defined for the ST.86 XML Schema, but MAY be done for implementation schemas.
66. The <any> element MUST not be used for basic components. It MAY be used for country-specific extension components and reference components.	66. The <any> element SHOULD be used to offer extension and to keep the ST.86 XML Schema (Appendix B) opened to additional elements. It MUST not be used in implementation schema.

67. Elements SHOULD be declared with occurrence indicators. The occurrence indicators should not be declared explicitly when the required value is the default value.	67. Elements SHOULD be declared with occurrence indicators. The occurrence indicators should not be declared explicitly when the required value is the default value. For example: <xs:element name="RegistrationNumber" type="xs:string" minOccurs="0"/> <xs:element name="DesignRepresentationSheet" type="RepresentationSheetType" maxOccurs="unbounded"/>
68. The content or value within tags and attributes may be in any language, except enumerations.	68. The content or value within tags and attributes may be in any language, except enumerations.
69. A revision history of schema SHOULD not be inserted in the schema itself. The reference to the revision history, and the latest version number and date of the schema SHOULD be only given in the XML schema. The revision history SHOULD document changes with a date and description of each change, in reverse chronological order and published via the Office's web site. For example:	69. A revision history of schema SHOULD not be inserted in the schema itself. The reference to the revision history, and the latest version number and date of the schema SHOULD be only given in the XML schema. The revision history SHOULD document changes with a date and description of each change, in reverse chronological order and published via the Office's web site. For example:
<xs:annotation>	<xs:annotation>
<xs:documentation>WIPO Standard XML4IP Model Schema Version 1.0, published in 2009-**-**. The revision history is available on WIPO website at http://www.wipo.int/standards/en/xml_material/XML4IP/revision-history/index.html </xs:documentation>	<xs:documentation>WIPO Standard ST.86 Model Schema Version 1.0, published in 2008-03-02. The revision history is available on WIPO website at http://www.wipo.int/standards/XMLSchema/designs/revision-history-st86model-schema.doc </xs:documentation>
</xs:annotation>	</xs:annotation>
70. Contact points SHOULD be included in the prologue.	70. Contact points SHOULD be included in the prologue. For example:
	<!-- Author : SDWG ST.86 Task Force -->

	<!-- Contact : xml.standards@wipo.int -->
Naming Office-Specific Types and Elements	**Naming Office-Specific Types and Elements**
71. Country-specific extension components are defined in individual namespaces. Namespace prefixes of country-specific extension components and basic components are arbitrary.	71. A namespace SHOULD be established for office-specific elements, where the office code (ST.3) becomes the prefix for identifying elements that are in that namespace. For example: xmlns:kr=" http://www.kipo.go.kr" <kr:element name="Bibliographic"> < kr:complexType mixed="true"> <kr:choice minOccurs="0" maxOccurs="unbounded"> <kr:element ref="DocumentCode"/> <kr:element ref="DocumentName" minOccurs="0"/> <kr:element ref="ReceiveOffice" minOccurs="0"/> <kr:element ref="ApplicationDate" minOccurs="0"/> </kr:complexType> </kr:element>
72. Names of office-specific types and elements that are not defined in the Standard MUST be distinguished using prefix of individual namespaces..	72. Types that are not defined in the Standard can be defined as office-specific. Type names SHOULD have a prefix specific to the organization followed by a single dash. In the case of industrial design offices, types SHOULD be prefixed by the two-letter office code as specified in WIPO Standard ST.3.

73. For organizations/offices not identified in WIPO Standard ST.3 or companies, they SHOULD not use two-letter codes but rather codes composed of three or four uppercase letters.	73. For organizations/offices not identified in WIPO Standard ST.3 or companies, they SHOULD not use two-letter codes but rather codes composed of three or four uppercase letters.
	74. As an alternative, a unique namespace SHOULD be established for office-specific elements, where the country code or company symbol becomes the prefix for identifying elements that are in that namespace.
	External Entities
	75. An external entity is any object that accompanies an XML document instance that is referenced from within the document instance. External entities are an integral part of an industrial design document. Without them, the XML instance cannot be parsed, rendered, or understood successfully.
76. External entities that are images of the mark SHALL conform to one or the other of the following profiles published by WIPO on its website.(2)	76. In the industrial design domain, an external entity is most frequently a view, usually of the representation of industrial designs. External entities that are views of the industrial designs SHALL conform to one or the other of the following profiles published by WIPO on its website. (2)
JPEG	JPEG
PNG	PNG
TIFF	TIFF
GIF	GIF
77. Images can be embedded in an XML document instance as embedded binary images encoded in base64Binary that is the W3C XML Schema standard data type as well as references to external image files, i.e., external entities. However images SHOULD be referenced as external entities.	77. Views can be embedded in a XML document instance as embedded binary images encoded in base64Binary that is the W3C XML Schema standard data type as well as references to external image files, i.e., external entities. However images SHOULD be referenced as external entities.