

СТАНДАРТ ST.96 – ДОДАТОК I**ПРАВИЛА ТА УГОДИ ПРОЕКТУВАННЯ З ВИКОРИСТАННЯМ XML**

Версія 2.2

*Редакція, схвалена Цільовою групою XML4IP
Комітету зі стандартів BOIV (КСВ) 17 серпня 2016 року*

Зміст

СТАНДАРТ ST.96 – ДОДАТОК I	1
1. ВСТУП	4
1.1 Огляд.....	4
1.2 Мета	4
1.3 Застосування документа	4
1.4 Структура документа	4
1.5 Терміни та позначення	5
1.5.1 Ключові слова.....	5
1.5.2 Загальні позначення	5
1.5.3 Ідентифікатори правил	6
2. УГОДИ ПРОЕКТУВАННЯ З ВИКОРИСТАННЯМ XML.....	6
2.1 Загальні правила проектування з використанням XML.....	6
2.2 Угоди про іменування в XML.....	6
2.2.1 Угоди про іменування структурних компонентів схем	7
2.2.2 Угоди про іменування файлів схем	9
2.3 Принцип модульної побудови.....	10
2.3.1 Модулі схем.....	10
2.3.2 Зовнішні посилання у схемах	11
2.4 Можливість повторного використання.....	11
2.5 Простори імен	12
2.5.1 Оголошення і кваліфікація просторів імен	13
2.5.2 Простори імен у схемах XML.....	13
2.5.3 Цільові простори імен.....	14
2.5.4 Простори імен за замовчуванням	15
2.6 Керування версіями схем.....	15

2.6.1	Значні та незначні зміни	16
2.6.1.1	Основні версії	16
2.6.1.2	Проміжні версії.....	16
2.6.2	Стратегія керування версіями схем	17
2.6.2.1	Угоди про іменування файлів і папок у межах керування версіями схем	17
2.6.2.2	Вбудований у схему XML атрибут <code>version</code> у межах керування версіями схем	17
2.6.2.3	Користувачькі атрибути <code>st96Version</code> та <code>ipoVersion</code> у межах керування версіями схем для екземплярів XML	18
2.7	Взаємоперетворення з іншими стандартами BOIV, що стосуються XML	20
2.8	Схеми промислових стандартів	21
3.	УГОДИ ПОБУДОВИ СХЕМ XML	22
3.1	Визначення типів	22
3.1.1	Прості типи.....	22
3.1.1.1	Вбудовані типи даних схеми W3C	22
3.1.1.2	Користувачькі типи даних	22
3.1.2	Комплексні типи.....	23
3.2	Елементи та атрибути	23
3.2.1	Елементи або атрибути	23
3.2.2	Елементи.....	24
3.2.2.1	Кардинальність елементів	24
3.2.2.2	Порожні елементи	24
3.2.3	Атрибути	24
3.2.4	Групування елементів та атрибутів	25
3.3	Розширення та обмеження.....	25
3.3.1	Розширення	25
3.3.2	Обмеження.....	26
3.3.3	Групи заміщення	26
3.4	Обмеження ідентифікації.....	26
3.5	Документування схем.....	27
3.5.1	Документування структурних компонентів схем	27
3.5.2	Документування заголовків схем	27
4.1	Простори імен в екземплярах XML	29
4.1.1	Валідація екземплярів XML	29

4.1.2	Оголошення просторів імен і кваліфікація в екземплярах документів XML.....	29
4.1.3	Простори імен екземплярів схеми W3C.....	31
4.1.4	Видимість просторів імен.....	31
4.2	Зовнішні сутності.....	32
ДОПОВНЕННЯ А – СПИСОК ПРАВИЛ ПРОЕКТУВАННЯ		33
Загальні правила проектування		33
Правила проектування схем		35
Правила проектування екземплярів		40
ДОПОВНЕННЯ В – ТЕРМІНИ ПРЕДСТАВЛЕННЯ.....		42
ДОПОВНЕННЯ С – ПЕРЕЛІК АКРОНІМІВ ТА АБРЕВІАТУР		44
ДОПОВНЕННЯ D – ПОСИЛАННЯ		47
Стандарти VOIB.....		47
Промислові стандарти		47

1. ВСТУП

1.1 Огляд

1. Цей стандарт призначений для забезпечення узгодженості всіх об'єктів промислової власності, визначених у стандарті BOIB ST.96, і спрощення процесу міжвідомчого обміну інформацією шляхом застосування схем XML, придатних для багаторазового використання і взаємодії між собою.

1.2 Мета

2. Мета цього додатка полягає у визначенні повного набору Правил та угод проектування для створення і використання схем та екземплярів XML, що охоплюють всі види інформації з промислової власності, для спрощення процесів подання, обробки, публікації та обміну даними у сфері промислової власності.

1.3 Застосування документа

3. Цей документ призначений для використання ВПВ, постачальниками даних з промислової власності, а також широким загалом, залученим до сфери промислової власності. Вони повинні використовувати цей документ як вихідні Правила та угоди проектування, яким мають відповідати усі схеми XML, аби вони вважалися сумісними зі стандартом BOIB ST.96. ВПВ можуть використовувати цей документ як настанову для розробки власних внутрішніх правил проектування. Перед тим, як розпочати розробку нових або модифікування вже існуючих схем XML, необхідно вивчити цей документ. Після завершення розробки схеми XML, цей документ має бути використаний для перевірки її відповідності правилам проектування.

1.4 Структура документа

4. Під час вивчення цього документа рекомендовано дотримуватися тієї послідовності, в якій він написаний. Цей документ складається з наступних розділів:

- Розділ 1 "Вступ" визначає загальні правила, що застосовуються у всьому документі;
- Розділ 2 "Угоди проектування з використанням XML" визначає загальні правила, що застосовуються до проектування як схем, так і екземплярів XML;
- Розділ 3 "Угоди побудови схем XML" визначає конкретні правила проектування з використанням специфікацій схеми W3C для створення інших схем XML;
- Розділ 4 "Правила проектування екземплярів XML" визначає конкретні правила проектування для створення екземплярів.

5. Крім того, цей документ включає чотири доповнення:

- Доповнення А "Огляд правил проектування", в якому узагальнено правила проектування, викладені в цьому документі;
- Доповнення В "Терміни представлення", що містить визначення термінів представлення;
- Доповнення С "Перелік акронімів та аббревіатур", що містить терміни та аббревіатури, які використовуються в цьому документі;
- Доповнення D "Посилання", що містить посилання на стандарти BOIV та інші промислові стандарти.

1.5 Терміни та позначення

1.5.1 Ключові слова

6. У звичайному значенні

- термін "схема XML" означає мову для описання структури й обумовлення змісту документів у форматі XML;
- термін "схема W3C" стосується схем XML, що повністю відповідають Рекомендаціям щодо мови визначення схеми W3C – *Схема XML. Частина 1. Структури і схема XML. Частина 2. Типи даних*;

7. У цьому документі

- термін "схема" стосується схеми XML, визначеної в Додатку III до стандарту BOIV ST.96;
- термін "компонент" стосується типу, елементу або атрибута.

8. Ключові слова "ПОВИНЕН", "НЕ ПОВИНЕН", "МАЄ", "НЕ МАЄ" і "МОЖЕ" у цьому документі повинні тлумачитися як описано в розділі "Терміни та визначення" стандарту BOIV ST.96. Ці слова, написані маленькими літерами, вживаються у звичайному значенні.

1.5.2 Загальні позначення

9. У цьому документі використовуються такі позначення:

- <>: позначає описовий термін покажчика місця заповнення, яке під час впровадження замінюється на певне значення;
- " ": вказує на те, що текст у лапках повинен відтворюватися під час упровадження дослівно;
- { }: позначає елементи, що є необов'язковими під час упровадження;

- шрифт *Courier*: позначає ключові слова XML, імена тегів XML і коди XML.

1.5.3 Ідентифікатори правил

10. Усі правила проектування є нормативними. Правила проектування ідентифікуються за допомогою префіксу [XX-nn], де

(a) "XX" є префіксом для розподілення правил за типами, а саме:

- "GD" для загальних правил проектування;
- "SD" для правил проектування схем;
- "ID" для правил проектування екземплярів.

(b) "nn" є наступним доступним у послідовності номером певного типу правил. Слід зазначити, що номер не вказує на позицію правила, а саме нового правила. Нове правило розміщується у відповідному контексті. Наприклад, ідентифікатор [GD-40] позначає сорокове загальне правило проектування. Правило [GD-40] може розміщуватися між правилами [GD-20] і [GD-21], замість того, щоб слідувати за правилом [GD-39], якщо таке розміщення є найбільш підходящим для цього правила.

(c) Ідентифікатор видаленого правила зберігається, тоді як власне правило замінюється на текст "Видалено".

2. УГОДИ ПРОЕКТУВАННЯ З ВИКОРИСТАННЯМ XML

2.1 Загальні правила проектування з використанням XML

11. Цей розділ містить основні загальні правила і настанови щодо проектування з використанням XML для всіх розробок, а не лише для певних аспектів використання XML. Загальні правила і настанови, викладені нижче, є основним підґрунтям для розробки даних і документів XML у сфері промислової власності.

- [GD-01] Усі схеми XML ПОВИННІ ґрунтуватися на технічних специфікаціях схеми W3C, що мають статус рекомендацій.
- [GD-02] Схеми ПОВИННІ відповідати вимогам таких документів:
Схема XML. Частина 1. Структури (<http://www.w3.org/TR/xmlschema-1>) і
Схема XML. Частина 2. Типи даних (<http://www.w3.org/TR/xmlschema-2>).
- [GD-03] Для символів ПОВИНЕН використовуватися стандарт [ISO/IEC 10646](http://www.iso.org/iso/10646) – Універсальний набір символів – *Unicode*. Символи *Unicode* ПОВИННІ кодуватися в UTF-8.

2.2 Угоди про іменування в XML

12. Угоди про іменування в XML є необхідними для забезпечення сумісності, однаковості та недвозначності імен і визначень усіх ресурсів XML.

2.2.1 Угоди про іменування структурних компонентів схем

13. Угоди про іменування в XML за стандартом BOIB ST.96 ґрунтуються на положеннях і настановах, викладених у частині 5 стандарту [ISO 11179](#). "Принципи іменування та ідентифікації". Імена типів, елементів та атрибутів складаються з таких термінів:

- Об'єктний клас (*Object Class*) стосується дії або об'єкта в контексті виробничої діяльності і представляє собою логічні групування або агрегування даних (в логічній моделі даних), до якої належить властивість. Об'єктний клас виражається терміном об'єктного класу;
- Властивість (*Property Term*) визначає характеристики об'єктного класу;
- Термін представлення (*Representation Term*) класифікує формат даних елементу з множини типів. Для цілей стандарту BOIB ST.96 мають використовуватися терміни представлення, наведені в Доповненні В до цього документа;
- Кваліфікатор (*Qualifier Term*) – це слово або слова, що допомагають визначати і відрізнити один елемент даних від інших, пов'язаних з ним елементами даних; за необхідності, він може бути приєднаний до об'єктного класу або властивості для створення унікального імені.

- [GD-04] Імена типів, елементів та атрибутів ПОВИННІ наводитися англійськими словами і відповідати вимогам до написання, зазначеним в Оксфордському словнику англійської мови (*Oxford English Dictionary*). Допустимими винятками є акроніми, аббревіатури та інші скорочення слів, наведені у Доповненні С.
- [GD-05] Імена типів, елементів та атрибутів МАЮТЬ складатися лише з іменників, прикметників і дієслів у теперішньому часі, за винятком акронімів, аббревіатур та інших скорочень слів, наведених у Доповненні С.
- [GD-06] Символи для використання в іменах типів, елементів або атрибутів ПОВИННІ знаходитися у межах такого набору: 'a-z, A-Z і 0-9'.
- [GD-07] Максимальна довжина імені компонента МАЄ складати не більше 35 символів.
- [GD-08] Імена типів, елементів та атрибутів МАЮТЬ бути стислими та очевидними.
- [GD-09] В іменах елементів ПОВИННА використовуватись угода upper camel case (UCC). Наприклад, `CountryCode`.
- [GD-10] В іменах типів ПОВИННА використовуватись угода UCC + суфікс "Type". Наприклад, `ApplicantType`.
- [GD-11] В іменах атрибутів ПОВИННА використовуватись угода lower camel case (LCC). Наприклад, `currencyCode="EUR"`.

- [GD-12] Акроніми та аббревіатури, наведені у Доповненні С, ПОВИННІ завжди використовуватися замість повних розширених імен.
- [GD-13] Акроніми та аббревіатури на початку оголошення атрибута ПОВИННІ наводитися у нижньому регістрі. Всі інші акроніми та аббревіатури в оголошенні атрибута ПОВИННІ наводитися у верхньому регістрі як зазначено у Доповненні С.
- [GD-14] Акроніми та аббревіатури для імен елементів і типів ПОВИННІ наводитися відповідно до Доповнення С.
- [GD-15] Імена комплексних типів МАЮТЬ містити значущі об'єктні класи.
- [GD-16] Якщо, зважаючи на вид суб'єкта господарювання, ім'я певного компонента має містити два терміни об'єктних класів, можна використовувати комплексний поєднаний тип. Ім'я комплексного поєданого типу МАЄ складатися з об'єктного класу комплексного поєднуючого типу, властивості, що описує характер поєднання, та об'єктного класу комплексного поєднуваного типу. Кваліфікатори можуть передувати об'єктному класу і властивості. Наприклад, в імені `ApplicantResidenceAddress: Applicant` є об'єктним класом комплексного поєднуючого типу, `Residence` є властивістю, а `Address` – об'єктним класом комплексного поєднуваного типу.
- [GD-17] Імена простих елементів (що базуються на простому або комплексному типі, який не допускає дочірніх елементів) МАЮТЬ складатися з об'єктного класу, властивості та, за наявності, терміну представлення. Кваліфікатори можуть передувати об'єктному класу та властивості.
- [GD-18] Об'єктний клас завжди ПОВИНЕН мати однакове семантичне значення у межах всього простору імен і МОЖЕ складатися з декількох слів. Наприклад, `ContactInformation`.
- [GD-19] Властивість в імені ПОВИННА бути унікальною в межах об'єктного класу, проте МОЖЕ використовуватися повторно в різних об'єктних класах.
- [GD-20] Кваліфікатор МОЖЕ, за необхідності, додаватися до об'єктного класу або властивості для індивідуалізації імені.
- [GD-21] Якщо ім'я містить об'єктний клас, властивість і термін представлення, об'єктний клас ПОВИНЕН передувати властивості, а властивість ПОВИННА передувати терміну представлення. Кваліфікатор МАЄ передувати поєднуваному об'єктному класу та властивості.
- [GD-22] Якщо властивість закінчується тим самим (або ж еквівалентним) словом, що і термін представлення, то термін представлення ПОВИНЕН бути видалений.
- [GD-23] Якщо потрібен термін представлення, терміни представлення, наведені у Доповненні В, ПОВИННІ використовуватися як терміни представлення в іменах базових компонентів.

- [GD-24] Усі імена типів, елементів та атрибутів у просторі імен ПОВИННІ бути унікальними.
- [GD-25] Слово або слова в імені МАЮТЬ наводитися в однині, за винятком, якщо поняття передається лише множиною. Наприклад, `TotalMarkSeries`.
- [GD-26] Ім'я елементу або типу, що містить сукупність контекстуально пов'язаних компонентів, МАЄ включати суфікс "Bag". Наприклад, `EmailAddressBag` означає сукупність елементів `EmailAddress`.
- [GD-27] Сполучні слова на зразок сполучника "and", прийменника "of" та артикля "the" НЕ МАЮТЬ використовуватися в іменах типів, елементів та атрибутів, за винятком, якщо вони складають частину виробничої термінології.
- [GD-28] Імена типів, елементів та атрибутів НЕ ПОВИННІ перекладатися, змінюватися або замінюватися за жодних умов.
- [GD-29] Імена типів та елементів НЕ ПОВИННІ містити посилання на номери статей і правил. Наприклад, `PCTRule702C`.

2.2.2 Угоди про іменування файлів схем

14. Назви файлів схем і назви схем зазвичай добираються парами. Назви файлів схем ґрунтуються власне на назвах схем. Наприклад, назва файла `PostalAddressType.xsd` базується на назві схеми `PostalAddressType`. Таким чином, угоди про іменування файлів схем співвідносяться з правилами іменування у межах XML у цьому документі.

15. Файл схеми МОЖЕ містити інформацію про її версію. Схема, що знаходиться на стадії проектування, може переглядатися. Проекти схем повинні позначатися шляхом додавання літери "D" і номеру редакції до їх назв.

- [GD-30] Символи для використання в іменах файлів схем ПОВИННІ знаходитися у межах такого набору: 'a-z, A-Z, 0-9, знак підкреслювання "_" і крапка ".".
- [GD-31] Назва файла схеми ПОВИННА складатися з двох обов'язкових частин з одним роздільником та необов'язкової інформації про її версію з двома додатковими роздільниками, тобто: <назва компонента>{"-"V"<номер основної версії>"-"<номер проміжної версії>"}."<розширення файла>. Наприклад, `EmailAddressType.xsd`, `languageCode.xsd`, `ApplicationBody_V1_0.xsd`.
- [GD-32] Назва файла проекту схеми ПОВИННА складатися з чотирьох обов'язкових частин з двома роздільниками та необов'язкової інформації про її версію з двома додатковими роздільниками, тобто: <назва компонента>{"_"V"<номер основної версії>"_"<номер проміжної версії>"}_"D"<номер редакції>"}."<розширення файла>, наприклад, `Contact_D3.xsd`, `TrademarkApplication_V1_1_D1`. Якщо проект схеми базується на існуючій схемі, а назва його файла містить інформацію про версію схеми, номери основної та проміжної версій у назві файла проекту схеми МАЮТЬ відповідати номерам, зазначеним у назві файла

схеми, на якій базується проект схеми. Якщо проект схеми є новим, номер основної версії у назві файла проекту схеми МАЄ відповідати номеру, визначеному в відповідному просторі імен, а номером проміжної версії у назві файла проекту схеми МАЄ бути "0".

2.3 Принцип модульної побудови

16. Стандарт BOIB ST.96 ґрунтується переважно на принципі модульної побудови під час проектування схем. Поділ схем на численні модулі забезпечує можливість імпортування повторно використовуваних компонентів замість повних схем. Тому стандарт BOIB ST.96 рекомендує уникати визначення всіх елементів і логічних компонентів в одній суцільній схемі XML, що зменшує можливість обміну та повторного використання окремих елементів і логічних компонентів, визначених у схемі як група.

2.3.1 Модулі схем

17. Компоненти, визначені у стандарті BOIB ST.96, поділяються за категоріями на компоненти типу *Common*, компоненти типу *Patent*, компоненти типу *Trademark* і компоненти типу *Design*. Компоненти типу *Common* МАЮТЬ бути нейтральними за контекстом (незалежними від виробничої діяльності) і використовуватись принаймні для двох об'єктів промислової власності. В окремих випадках, зважаючи на розвиток стандарту BOIB ST.96 у майбутньому, окремі компоненти, наприклад "P", можуть визначатися як загальні.

18. Існують три типи компонентів: основний (*Basic*), сукупний (*Aggregate*) і документ (*Document*); компоненти типу *Document*, що складаються з компонентів типу *Basic*, *Aggregate*, та/або інші компоненти типу *Document*, визначаються з урахуванням виробничих цілей. Компоненти типу *Basic* та *Aggregate* визначаються в області *Common*. Компоненти типу *Basic*, *Aggregate* і *Document* визначаються в області *Patent*, *Trademark* і *Design*.

Рівень	Опис
Компонент типу <i>Basic</i>	Компонент типу <i>Basic</i> означає вбудований тип даних схеми XML W3C, простий або комплексний тип з визначенням <code>xsd:simpleContent</code> . Він представляється елементом або атрибутом. Наприклад, <code>FirstName (string)</code> , <code>PhoneNumberCategory (PhoneNumberCategoryType)</code> .
Компонент типу <i>Aggregate</i>	Компонент типу <i>Aggregate</i> означає сукупність пов'язаних компонентів типу <i>Basic</i> та/або компонентів типу <i>Aggregate</i> , які разом передають очевидне виробниче значення, що не залежить від або не відноситься до будь-якого контексту виробничої діяльності. Він представляється елементом або комплексним типом. Наприклад, <code>Name (NameType)</code> , <code>Contact (ContactType)</code> .

Компонент типу <i>Document</i>	Компонент типу <i>Document</i> означає групу компонентів типу <i>Basic</i> , <i>Aggregate</i> та/або інших компонентів типу <i>Document</i> , який представляє певний аспект діяльності у сфері промислової власності, наприклад подання заявки, експертизу, публікацію, видачу охоронного документа та обмін даними. Наприклад, <i>ApplicationBody</i> , що ґрунтується на <i>ApplicationBodyType</i> , представляє опис до патенту і може використовуватись для подання заявки та публікації опису до патенту.
--------------------------------	--

2.3.2 Зовнішні посилання у схемах

19. Кожний модуль схеми знаходиться у певному просторі імен. У стандарті VOIB ST.96 кожний модуль схеми визначається в одному з чотирьох просторів імен, тобто, для модулів схем компонента типу *Common*, для модулів схем компонента типу *Patent*, модулів схем компонента типу *Trademark* і для модулів схем компонента типу *Design*.

20. Для зовнішніх посилань у схемі використовуються два структурних компоненти схеми W3C, а саме, *include* та *import*. Структурний компонент *include* необхідно використовувати, якщо схема, яка включає, і схема, яка включається, знаходяться в одному заданому просторі імен. Термін "схема, яка включає" означає схему, яка включає зовнішню схему, а термін "схема, яка включається" означає власне зовнішню схему. Структурний компонент *import* необхідно використовувати, якщо схема, яка включає, і схема, яка включається, знаходяться у різних просторах імен. Цей метод є особливо корисним, якщо відомству не потрібно обмежувати використання у схемах лише тих структурних компонентів, які належать до певного простору імен.

- [SD-01] Модулі схем для компонента *Patent*, модулі схем для компонента *Trademark* і модулі схем для компонента *Design* ПОВИННІ використовувати структурний компонент *xsd:import* для посилання на модулі схем компонента типу *Common*.
- [SD-02] Модулі схем компонента типу *Common* НЕ ПОВИННІ посилатися на модулі схем компонента типу *Patent*, модулі схем компонента типу *Trademark* або модулі схем компонента типу *Design*.
- [SD-03] Модуль схеми компонента типу *Patent*, *Trademark* або *Design* МОЖЕ посилатися на модулі схем компонента типу *Common*, на схеми у тому ж модулі компонента типу *Common* і на погоджені схеми за промисловими стандартами, проте він НЕ ПОВИНЕН посилатися на інші модулі схем компонента типу *Common*. Наприклад, схеми компонента типу *Patent* НЕ ПОВИННІ посилатися на схеми компонента типу *Trademark* і навпаки.

2.4 Можливість повторного використання

21. Важливими принципами побудови схем XML є поділ і поєднання. Поділ має на меті мінімізувати залежність елементів один від одного, як в екземплярі XML, так і в схемі.

Поєднання призначене для групування пов'язаних частин даних. Деякі проектні шаблони виникли внаслідок такого адресного поділу і поєднання схем XML.

22. Проектні шаблони також дають відповідь на питання регулювання ступеня деталізації компонентів (елементів і типів) для повторного використання. Вибір належного шаблону є важливим кроком на етапі проектування схем. Таким чином, проектний шаблон схеми слід вибирати до початку її побудови.

23. Найбільш розповсюдженими шаблонами проектування є *"Russian Doll"* (матрьошка), *"Salami Slice"* (шматок салями), *"Venetian Blind"* (жалюзі), а також *"Garden of Eden"* (Едем). Шаблони відрізняються кількістю глобальних компонентів (елементів або типів). Для розуміння шаблону проектування необхідно розрізнити глобальний компонент від локального. Глобальний компонент – є підрядною складовою першого рівня структурного компонента схеми у файлі визначення схеми XML. Локальний компонент не є підрядною складовою першого рівня структурного компонента схеми у файлі визначення схеми XML. Глобальні компоненти пов'язані із заданим простором імен схеми і можуть бути використані в іншій схемі. Також слід пам'ятати, що будь-який елемент, визначений у глобальному просторі імен, може бути кореневим для правильного екземпляра XML, поєданого зі схемою, визначеною для цього простору імен.

24. У більшості сучасних схем використовуються шаблони *"Venetian Blind"* або *"Garden of Eden"* через їх значні можливості повторного використання. З точки зору поділу та поєднання, шаблон *"Venetian Blind"* є кращим за шаблон *"Garden of Eden"*, але з точки зору повторного використання, перевага на боці *"Garden of Eden"* через використання як елементів, так і глобальних типів. Найбільшою проблемою підходу до проектування, заснованого на типах (*"Venetian Blind"*), є ризик формування несталого словника елементів, у якому елементи оголошуються локально і можуть використовуватися повторно, незалежно від семантичної ясності та узгодженості типів. Тому, стандарт BOIB ST.96 рекомендує використовувати шаблон проектування *"Garden of Eden"*.

- [SD-04] Якщо можливо, ПОВИННІ використовуватися існуючі схеми, а не створюватися нові.
- [SD-05] Схеми МАЮТЬ максимально використовувати елементи і типи, визначені в існуючих схемах.
- [SD-06] Усі типи, елементи та атрибути ПОВИННІ оголошуватися глобально.
- [SD-07] Схеми НЕ ПОВИННІ використовувати структурний компонент `xsd:redefine`.

2.5 Простори імен

25. Простори імен використовують для однозначної ідентифікації елементів та атрибутів з одним і тим самим іменем у випадку їх поєднання в одному документі. Простори імен встановлюють зв'язок структурних компонентів схеми з концептуальним простором, що визначає словник розмітки. Екземпляр XML може містити імена елементів та атрибутів з

декількох словника XML. Якщо кожному з цих словників присвоєно окремий простір імен, можна уникнути неоднозначності між ідентично названими елементами або атрибутами. Простори імен повинні бути унікальними і сталими. Стандарт BOIB ST.96 рекомендує використовувати конфігурацію з декількома просторами імен.

- [SD-08] Схеми ПОВИННІ використовувати простори імен.
- [SD-09] Опубліковані оголошення просторів імен НЕ МАЮТЬ змінюватися.

2.5.1 Оголошення і кваліфікація просторів імен

- [SD-10] Схеми ПОВИННІ містити оголошення простору імен схеми W3C.
- [SD-11] Схеми ПОВИННІ використовувати кваліфікації просторів імен для всіх структурних компонентів схеми W3C.
- [SD-12] Схеми ПОВИННІ використовувати простір імен у форматі URI.
- [SD-13] Усі схеми ПОВИННІ мати атрибути `elementFormDefault` і `attributeFormDefault` зі значенням "qualified" у кореневому елементі `xsd:schema`.

26. Для забезпечення ефективності, ВПВ можуть зменшити або взагалі відмовитися від використання просторів імен у виробничих системах обробки за допомогою XML. Проте, цих правил необхідно дотримуватися під час обміну інформацією.

2.5.2 Простори імен у схемах XML

27. Стандарт BOIB ST.96 визначає чотири простори імен: *Common*, *Patent*, *Trademark* і *Design*. Ці простори імен є символьними і постійними в усіх випусках.

28. Наступний приклад демонструє належно сформований простір імен:

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

- [SD-14] Назви просторів імен ПОВИННІ мати таку структуру: <http://www.wipo.int/standards/XMLSchema/<category>>, де `<category>` – це ідентифікатор використовуваного домену схеми: *Common*, *Patent*, *Trademark* або *Design*.
- [SD-15] Схеми МАЮТЬ використовувати "xsd" як префікс простору імен для всіх структурних компонентів схеми W3C, "com" для всіх схем компонента типу *Common*, "pat" для всіх схем компонента типу *Patent*, "tmk" для всіх схем компонента типу *Trademark*, "dgn" для всіх схем компонента

типу *Design*, "tbl" для схем табличного компонента *OASIS* і "mathml" для схем *MathML*.

2.5.3 Цільові простори імен

29. Оголошення цільового простору імен у схемі забезпечує об'єднання всіх структурних компонентів схеми у межах одного простору імен. І навпаки, без заданого простору імен структурні компоненти, оголошені у схемі, не належатимуть до жодного простору імен. Хоча схема може мати декілька оголошених просторів імен, лише один з них може бути заданим. Рекомендації W3C не вимагають, аби заданий простір імен був оголошений у схемі. Проте, стандарт BOIB ST.96 рекомендує використовувати заданий простір імен.

30. Схема містить оголошення заданого простору імен, що відповідає одному з оголошених просторів імен. Відповідно до пов'язаного простору імен, схеми повинні містити оголошення одного з наступних чотирьох заданих просторів імен, визначених у стандарті BOIB ST.96:

- <http://www.wipo.int/standards/XMLSchema/Common>, наприклад,

```
<?xml version="1.0" UTF-8?>
<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:com="http://www.wipo.int/standards/XMLSchema/Common"
targetNamespace="http://www.wipo.int/standards/XMLSchema/Common"
elementFormDefault="qualified" attributeFormDefault="qualified" version="V2_0">
...
</xsd:schema>
```

- <http://www.wipo.int/standards/XMLSchema/Patent>, наприклад,

```
<?xml version="1.0" UTF-8?>
<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:pat="http://www.wipo.int/standards/XMLSchema/ST96/Patent"
targetNamespace="http://www.wipo.int/standards/XMLSchema/ST96/Patent"
elementFormDefault="qualified" attributeFormDefault="qualified" version="V2_0">
<xsd:include schemaLocation="xxx.xsd"/>
</xsd:schema>
```

- <http://www.wipo.int/standards/XMLSchema/Trademark>, наприклад,

```
<?xml version="1.0" UTF-8?>
<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:tmk="http://www.wipo.int/standards/XMLSchema/ST96/Trademark"
targetNamespace="http://www.wipo.int/standards/XMLSchema/ST96/Trademark"
elementFormDefault="qualified" attributeFormDefault="qualified" version="V2_0">
...
</xsd:schema>
```

- <http://www.wipo.int/standards/XMLSchema/Design>, наприклад,

```
<?xml version="1.0" UTF-8?>
<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:dgn="http://www.wipo.int/standards/XMLSchema/ST96/Design"
targetNamespace="http://www.wipo.int/standards/XMLSchema/ST96/Design"
elementFormDefault="qualified" attributeFormDefault="qualified" version="V2_0">
...
</xsd:schema>
```

- [SD-16] Кожна схема **ПОВИННА** містити оголошення цільового простору імен з використанням атрибута `xsd:targetNamespace`.
- [SD-17] Атрибут схеми `targetNamespace` **ПОВИНЕН** відповідати назві одного з оголошених просторів імен, окрім простору імен схеми W3C.
- [SD-18] Схеми компонента типу *Common* **ПОВИННІ** використовувати <http://www.wipo.int/standards/XMLSchema/Common> як цільовий простір імен.
- [SD-19] Схеми компонента типу *Patent* **ПОВИННІ** використовувати <http://www.wipo.int/standards/XMLSchema/Patent> як цільовий простір імен.
- [SD-20] Схеми компонента типу *Trademark* **ПОВИННІ** використовувати <http://www.wipo.int/standards/XMLSchema/Trademark> як цільовий простір імен.
- [SD-21] Схеми компонента типу *Design* **ПОВИННІ** використовувати <http://www.wipo.int/standards/XMLSchema/Design> як цільовий простір імен.

2.5.4 Простори імен за замовчуванням

31. Простір імен за замовчуванням зменшує надмірний рівень деталізації у схемі. Однак, оголошення простору імен за замовчуванням підвищує рівень неоднозначності, тому що вилучення префіксів ускладнює ідентифікацію простору імен, до якого належить структурний компонент. Використання просторів імен за замовчуванням може викликати перетворення типів, унеможливаючи визначення походження структурних компонентів включених до схеми шляхом їх перевірки. Тому стандарт BOIB ST.96 не рекомендує використовувати простір імен за замовчуванням.

- [SD-22] Схеми **НЕ МАЮТЬ** використовувати простір імен за замовчуванням.

2.6 Керування версіями схем

32. Якщо до існуючих схем потрібно внести зміни, основним питанням є зворотна сумісність. Зазвичай зворотна сумісність означає, що екземпляр XML, який відповідав попередній версії схеми, відповідає також і новій версії схеми.

2.6.1 Значні та незначні зміни

33. Зміни, що вносяться до схем, поділяються на значні та незначні. Значні зміни – це зміни, що призводять до порушення сумісності з попередніми версіями схеми, а незначні зміни є зворотно сумісними. Нові проміжні версії схем (з незначними змінами) **ПОВИННІ** забезпечувати відповідність екземплярів, створених на підставі попередніх проміжних версій цих схем (з незначними змінами), належній основній версії (зі значними змінами). Проте, не слід очікувати, що екземпляри відповідатимуть версіям схеми, що передували версії, на підставі якої вони були створені. Крім того, дозволяється проводити перегляд схеми на етапі її проектування.

2.6.1.1 Основні версії

34. Основна версія схеми охоплює значні зміни та/або зміни, що не є зворотно сумісними. У випадку перевірки будь-якого подібного екземпляра XML, що ґрунтується на основній версії, на відповідність новій версії схеми, можуть виникнути помилки перевірки. Нова основна версія схеми створюється у разі внесення значних змін та/або змін, що не є зворотно сумісними, наприклад у випадку:

- вилучення значень з переліків або зміна значень у них;
- змінювання назв елементів, типів або атрибутів;
- видалення або додавання обов'язкових елементів чи атрибутів;
- змінювання множини, тобто необов'язкової на обов'язкову.

2.6.1.2 Проміжні версії

35. Відповідно до стандарту BOIB ST.96 проміжна версія охоплює низку незначних або зворотно сумісних змін. Присвоєння схемі за стандартом BOIB ST.96 статусу проміжної означає, що вона є зворотно сумісною з іншою схемою за стандартом BOIB ST.96 попередньої або наступної проміжної версії у межах однієї основної версії. Таким чином, проміжні версії сприяють забезпеченню зворотної або прямої сумісності. Номер проміжної версії збільшується лише у разі внесення сумісних змін. Наприклад:

- додавання значень до переліків,
- необов'язкове розширення,
- додавання необов'язкових елементів та/або атрибутів.

[SD-23] Нові номери проміжних версій схем **ПОВИННІ** забезпечувати відповідність усіх екземплярів, створених на підставі попередніх проміжних версій схем, одній основній версії.

[SD-24] Якщо нова схема не забезпечує відповідність існуючого екземпляра XML, створеного на підставі поточної основної версії, номер основної версії схеми **ПОВИНЕН** бути збільшений.

- [SD-25] *Видалено*
- [SD-26] *Видалено*
- [SD-27] У випадку оновлення включеної або імпортованої схеми, номер основної та/або проміжної версії схеми МАЄ бути змінений.
- [SD-28] Якщо будь-який структурний компонент схеми зазнав змін, номери версій усіх схем у випуску ПОВИННІ бути збільшені.
- [SD-29] Під час створення нової схеми МАЮТЬ використовуватися включені або імпортовані схеми останніх версій.

2.6.2 Стратегія керування версіями схем

36. Визначення стратегії керування версіями схем передбачає одночасне врахування наступних угод:

- a) угод про іменування файлів схем і структури папок у випуску схем;
- b) використання вбудованого атрибута `version` схеми XML;
- c) використання користувацького атрибута `schemaVersion` на рівні екземпляра документа
- d) використання користувацького атрибута `ipoVersion` на рівні екземпляра документа.

2.6.2.1 Угоди про іменування файлів і папок у межах керування версіями схем

37. Інформація про версію МАЄ включатися до імені папки, до імені файла схеми на рівні документа і до імені файла зведеної схеми. Відповідно до правил іменування файлів, визначених у цьому документі, ім'я папки з випуском схеми, ім'я файла схеми на рівні документа та ім'я файла зведеної схеми містять номер як основної версії, так і проміжної. У структурі папок випуску схеми відповідно до стандарту BOIB ST.96 за номером версії випуску слідує тип компонента, тобто, *Common*, *Patent*, *Trademark*, *Design* і *External Standards*. Наприклад, версія 2.0 компонента *Contact* розміщується у папці *ST96/V2_0/Common*.

- [SD-64] Папка випуску схеми, ім'я файла схеми на рівні документа та ім'я файла зведеної схеми ПОВИННІ містити відповідну інформацію про версію, що складається з номера основної версії та номера проміжної версії.

2.6.2.2 Вбудований у схему XML атрибут `version` у межах керування версіями схем

38. Специфікація схеми W3C дозволяє визначити атрибут `version` у корінному елементі схеми. Відповідно до стандарту BOIB ST.96, схеми повинні включати як основний, так і додатковий номер версії для кожного файла схеми, що використовує атрибут `version` схеми W3C.

Наприклад,

```
<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:pat="http://www.wipo.int/standards/XMLSchema/ST96/Patent"
targetNamespace="http://www.wipo.int/standards/XMLSchema/ST96/Patent"
elementFormDefault="qualified" attributeFormDefault="unqualified" version="
V2_0">
...
</xsd:schema>
```

[SD-30] Атрибут `version` схеми W3C ПОВИНЕН складатися з номера основної версії та номера проміжної версії для кожного файлу схеми у такому форматі: "V"<номер основної версії>"_"<номер проміжної версії>.

2.6.2.3 Користувацькі атрибути `st96Version` та `ipoVersion` у межах керування версіями схем для екземплярів XML

39. Користувацький атрибут `st96Version` необхідний для того, щоб переконатися, що даний екземпляр документа відноситься безпосередньо до конкретної версії схеми XML за стандартом BOIB ST.96. Цей атрибут чітко визначає версію схеми за стандартом BOIB ST.96, на яку рівняється екземпляр документа або на яку посилається запроваджена схема XML. Цей атрибут має оголошуватися як необов'язковий і не поширюваний, коли компоненти схеми за стандартом BOIB ST.96 використовуються без змін.

40. Користувацький атрибут `ipoVersion` необхідний для того, щоб переконатися, що даний екземпляр документа відноситься безпосередньо до конкретної версії запровадженої відомчої схеми. Цей атрибут чітко визначає версію відомчої схеми, на яку рівняється екземпляр документа. Цей атрибут має оголошуватися як необов'язковий і не поширюваний, коли компоненти схеми за стандартом BOIB ST.96 використовуються без змін.

41. Зважаючи на зворотну і пряму сумісність, а також обмін даними між ВПВ, у межах якого використовуються різні проміжні версії, необхідно наводити номери основної і проміжної версій. У наступних прикладах показано як атрибути можуть бути визначені у схемі XML за стандартом BOIB ST.96, у відомчих схемах і зразках XML.

42. У наступних прикладах показано як атрибути `st96Version` та `ipoVersion` мають визначатися у схемі за стандартом BOIB ST.96 і відомчій схемі, а також використовуватись у зразках XML:

атрибут `st96Version` за стандартом BOIB ST.96 версії 2.0:

```
<?xml version="1.0" encoding="UTF-8"?>
<xsd:schema xmlns:com="http://www.wipo.int/standards/XMLSchema/ST96/Common"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
targetNamespace="http://www.wipo.int/standards/XMLSchema/ST96/Common"
elementFormDefault="qualified" attributeFormDefault="qualified" version="V2_0">
<xsd:attribute name="st96Version" type="xsd:token" fixed="V2_0">
...

```

```
</xsd:attribute>
</xsd:schema>
```

атрибут ipoVersion за стандартом BOIB ST.96 версії 2.0:

```
<?xml version="1.0" encoding="UTF-8"?>
<xsd:schema xmlns:com="http://www.wipo.int/standards/XMLSchema/ST96/Common"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
targetNamespace="http://www.wipo.int/standards/XMLSchema/ST96/Common"
elementFormDefault="qualified" attributeFormDefault="qualified" version="V2_0">
<xsd:attribute name="ipoVersion" type="xsd:token">
...
</xsd:att
```

атрибут ApplicationBodyType за стандартом BOIB ST.96 версії 2.0:

```
<?xml version="1.0"?>
<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema"
targetNamespace="http://www.wipo.int/standards/XMLSchema/ST96/Patent"
xmlns:com="http://www.wipo.int/standards/XMLSchema/ST96/Common"
xmlns:pat="http://www.wipo.int/standards/XMLSchema/ST96/Patent"
elementFormDefault="qualified" attributeFormDefault="qualified" version="V2_0">
<xsd:import namespace="http://www.wipo.int/standards/XMLSchema/ST96/Common"
schemaLocation="../../../Common/st96Version.xsd"/>
<xsd:import namespace="http://www.wipo.int/standards/XMLSchema/ST96/Common"
schemaLocation="../../../Common/ipoVersion.xsd"/>
.....
<xsd:complexType name="ApplicationBodyType">
.....
<xsd:attribute ref="com:st96Version" use="required" />
<xsd:attribute ref="com:ipoVersion"/>
.....
</xsd:complexType>
</xsd:schema>
```

атрибути ipoVersion та ApplicationBodyType за відомчою схемою Відомства США з патентів і торговельних марок (USPTO) версії 1.0, що базуються на атрибуті ApplicationBodyType за стандартом BOIB ST.96 версії 2.0:

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

```
<?xml version="1.0"?>
<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:uspat="urn:us:gov:doc:uspto:patent"
targetNamespace="urn:us:gov:doc:uspto:patent"
xmlns:com="http://www.wipo.int/standards/XMLSchema/ST96/Common"
```

```

xmlns:pat="http://www.wipo.int/standards/XMLSchema/ST96/Patent"
elementFormDefault="qualified" attributeFormDefault="qualified" version="V2_0">
<xsd:import namespace="http://www.wipo.int/standards/XMLSchema/ST96/Common"
schemaLocation="../../Common/st96Version.xsd"/>
<xsd:import namespace="http://www.wipo.int/standards/XMLSchema/ST96/Common"
schemaLocation="../../Common/ipoVersion.xsd"/>
.....
<xsd:complexType name="ApplicationBodyType">
.....
<xsd:attribute ref="com:st96Version" use="required" />
<xsd:attribute ref="com:ipoVersion" />
.....
</xsd:complexType>
</xsd:schema>

```

Зразок XML, що базується на елементі ApplicationBody за відомчою схемою USPTO версії 1.0:

```

<?xml version="1.0" encoding="UTF-8"?>
<uspat: ApplicationBody xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:com="http://www.wipo.int/standards/XMLSchema/ST96/Common"
xmlns:pat="http://www.wipo.int/standards/XMLSchema/ST96/Patent"
xmlns:uspat="urn:us:gov:doc:uspto:patent"
xmlns:uscom="urn:us:gov:doc:uspto:common" "
xsi:schemaLocation="urn:us:gov:doc:uspto:patent
PE2E/V1_0/USPatent/Document/ApplicationBody_V1_0.xsd" com:st96Version="V2_0"
com:ipoVersion="US_V1_0">
.....
</uspat:ApplicationBody>

```

[SD-31] Схема для компонента Document **ПОВИННА** визначати обов'язковий атрибут з іменем `st96Version` у своєму кореновому елементі з постійним значенням, що відповідає версії випуску схеми XML за стандартом BOIB ST.96 у такому форматі: "V"<номер основної версії>"_ "<номер проміжної версії>.

[SD-63] Схема для компонента Document **ПОВИННА** оголошувати необов'язковий атрибут `ipoVersion` у своєму кореновому елементі з постійним значенням, що відповідає версії випуску відомчої схеми у такому форматі: <ST.3 Code>"_ ""V"<номер основної версії>"_ "<номер проміжної версії>, наприклад "US_V2_0".

2.7 Взаємоперетворення з іншими стандартами BOIB, що стосуються XML

43. До прийняття стандарту BOIB ST.96, ВПВ користувалися стандартами BOIB [ST.36](#), [ST.66](#) і [ST.86](#), тому підтримка сумісності зі стандартами BOIB [ST.36](#), [ST.66](#) і [ST.86](#) є одним із головних завдань стандарту BOIB ST.96. Стандарт BOIB ST.96 намагається визначити необхідний рівень сумісності та взаємоперетворення зі стандартами BOIB [ST.36](#), [ST.66](#) або [ST.86](#) з метою забезпечення можливості прийнятної обробки даних для виробничих потреб ВПВ і постачальників інформації з промислової власності.

44. Взаємоперетворення означає, що низка атомарних інформаційних одиниць (найнижчий рівень деталізації елементів та атрибутів, зазвичай елемент без дочірніх

елементів), що може бути включена до екземпляра XML, сумісного зі стандартом BOIB ST.96, є взаємоперетворюваною з атомарними інформаційними одиницями, що можуть бути включені до екземпляру XML, сумісного зі стандартом BOIB [ST.36](#), [ST.66](#) або [ST.86](#), і навпаки.

45. Переліки значень мають містити низку допустимих значень для відповідних елементарних одиниць інформації, дозволених відповідно до стандарту BOIB [ST.36](#), [ST.66](#) або [ST.86](#). За необхідності, переліки значення мають включати значення "Other" (*інше*), що означає відсутність у списку значення, наявного у початкових даних, і "Undefined" (*не визначено*), що означає відсутність інформації у початкових даних (екземплярах XML, що відповідають стандартам BOIB [ST.36](#), [ST.66](#) і [ST.86](#)). З історичних причин, у деяких переліках значень використовується значення "Undefined" замість значення "Unknown".

[SD-32] Схеми за стандартом BOIB ST.96 МАЮТЬ розроблюватися з однаковим рівнем кардинальності¹ та деталізації для спрощення взаємоперетворення екземплярів XML за стандартом BOIB ST.96 на екземпляри XML за стандартом BOIB [ST.36](#), [ST.66](#) або [ST.86](#).

[SD-33] Переліки значень у схемах НЕ МАЮТЬ включати значення "Other" або "Unknown", якщо відсутня необхідність.

2.8 Схеми промислових стандартів

46. Якщо це вимагається змістом документа, тобто, якщо зміст не є унікальним для сфери промислової власності, мають використовуватись схеми промислових стандартів. Схемами промислових стандартів, рекомендованими цим документом, є таблична схема XML OASIS і схема MathML.

47. Таблична схема XML OASIS, доступна за адресою: <http://www.oasis-open.org/docbook/xmlschema/1.0b1/calstbl.xsd>, є типовою схемою і потребує доопрацювання залежно від потреб виробничої діяльності. Схема OASISTable.xsd визначається у межах схеми за стандартом BOIB ST.96 на основі табличної схеми OASIS.

48. Для математичних формул рекомендовано використовувати схему MathML версії 3. Докладна інформація стосовно схеми MathML версії 3 доступна за адресою: <http://www.w3.org/TR/MathML3>.

49. На момент підготовки цього документа не існувало жодних рекомендацій W3C або погодженого промислового стандарту стосовно XML для хімічних формул або структур. Хоча для різних хімічних типів існує велика кількість стандартів стосовно XML, жоден з них не вважається загальноприйнятим. Тому цей документ не містить жодних рекомендацій щодо стандарту XML для хімічних даних.

[SD-34] Для математичних формул МАЄ використовуватись схема MathML версії 3.

¹ Тут "кардинальність" означає кількість елементів у наборі.

[SD-35] Для таблиць МАЄ використовуватись схема OASISTable.xsd, створена на основі схеми *OASIS Exchange Table*.

[SD-36] Схеми промислових стандартів МАЮТЬ включатися лише за допомогою посилання і після попереднього погодження Комітету зі стандартів BOIB.

3. УГОДИ ПОБУДОВИ СХЕМ XML

3.1 Визначення типів

50. Типи представляють собою види інформації, які можуть містити елементи та атрибути, наприклад рядки символів або дати. Типи та елементи зазвичай добираються парами.

3.1.1 Прості типи

51. Використання простих типів сприяє підвищенню якості даних у застосуваннях XML через те, що всі застосування, які використовують прості типи, підлягають одним і тим самим перевіркам обробниками XML. Якщо лексична форма простого типу не є прийнятною, розробники схем можуть створювати власні типи даних, використовуючи синтаксис регулярних виразів схеми W3C. Прості типи включають як вбудовані типи даних схеми W3C, так і користувацькі типи даних. Стандарт BOIB ST.96 визначає прості типи як користувацькі типи даних.

3.1.1.1 Вбудовані типи даних схеми W3C

52. Вбудовані типи даних схеми W3C – це типи даних, визначені W3C, включені до стандарту схеми W3C, наприклад, `date`, `Boolean`, `string` і `token`.

3.1.1.2 Користувацькі типи даних

53. Однією з переваг схем XML є можливість створення користувацьких типів даних. Користувацькі типи даних є типами даних, заснованими на існуючих вбудованих типах даних схеми W3C, які також можуть походити з уже існуючих користувацьких типів даних. Користувацькі типи даних можуть бути отримані одним із трьох способів: `restriction` (обмеження), `list` (перелічення) і `union` (об'єднання).

[SD-37] Схеми МАЮТЬ максимально використовувати прості типи.

[SD-38] Списки кодів МАЮТЬ оголошуватися як переліки значень у простому типі.

[SD-39] Список кодів МОЖЕ бути оголошений як поєднання простих типів.

[SD-40] Для представлення ВПВ, пріоритетних і зазначених країн/організацій ПОВИННІ використовуватися двобуквені коди за стандартом BOIB [ST.3](#). Наприклад, `PriorityCountryCode="EP"`.

[SD-41] Для представлення назв країн, колоній та інших територій, що представляють певний геополітичний інтерес, ПОВИННІ

використовуватися кодувальні елементи Альфа-2 за стандартом [ISO 3166-1](#) (двобуквені коди назв країн) на основі списків назв країн, отриманих від Організації Об'єднаних Націй.

- [SD-42] Для кодування мов ПОВИННІ використовуватися коди за стандартом [ISO 639-1](#) (двобуквені коди мов).
- [SD-43] Схеми ПОВИННІ оголошувати елементи та атрибути для дати і часу з використанням типів даних часу і дати за схемою W3C.
- [SD-44] Для кодування валют ПОВИННІ використовуватися коди за стандартом [ISO 4217](#) (трибуквені коди валют).
- [SD-45] Символи, що використовуються у переліках значень, ПОВИННІ обмежуватися таким набором: 'a-z, A-Z, 0-9, пробіл " " і знак підкреслювання "_". Перелічувані значення НЕ МАЮТЬ починатися з цифрового символу.
- [SD-46] Перелічувані значення МАЮТЬ бути "семантично достатніми" і наводитися англійською мовою якомога меншою кількістю символів. Значення МАЮТЬ походити із загальноприйнятої термінології у сфері промислової власності.

3.1.2 Комплексні типи

54. Комплексні типи – це користувацькі типи, що містять дочірні елементи та/або атрибути.

- [SD-47] МОЖУТЬ використовуватися абстрактні комплексні типи.

3.2 Елементи та атрибути

55. Елементи є основними структурними компонентами екземпляра XML і представляються за допомогою тегів. Атрибути є структурними компонентами схеми W3C, пов'язаними з елементами, що надають додаткову інформацію стосовно елементів.

3.2.1 Елементи або атрибути

56. Одним із ключових питань під час проектування схем є питання представлення інформаційних об'єктів як елементів або атрибутів XML. Тоді як елементи вважаються такими, що містять дані, атрибути вважаються такими, що містять метадані. Якщо інформаційний об'єкт був представлений як атрибут, його подальше розширення неможливе і, таким чином, атрибути МАЮТЬ використовуватися лише для опису інформації, яку надалі не можна або яку не планується розширювати або деталізувати. Схеми МАЮТЬ проектуватися таким чином, аби елементи були основними контейнерами інформації з промислової власності в екземплярах XML. Атрибути МАЮТЬ містити допоміжні метадані, тобто прості фрагменти з додатковою інформацією про елементи.

- [SD-48] Схеми МАЮТЬ використовувати атрибути лише для визначення невірних даних. Наприклад, для номерів послідовностей допустимо використовувати атрибут `sequenceNumber`.

3.2.2 Елементи

3.2.2.1 Кардинальність елементів

57. Термін "кардинальність" визначається як кількість елементів у наборі.
58. Кардинальність позначається у схемі за допомогою обмежувачів `minOccurs` і `maxOccurs` в оголошенні елементу; ці обмежувачі також відомі як показники появи. Показники появи не можуть зазначатися у глобальних оголошеннях елементів.

- [SD-49] Показник появи НЕ МАЄ використовуватись для зазначення обмеження, яке є стандартним у межах схеми. Наприклад, `minOccurs="1"` і `maxOccurs="1"` НЕ МАЮТЬ використовуватись, проте `minOccurs="2"` і `maxOccurs="3"` є прийнятними.

3.2.2.2 Порожні елементи

59. Порожній елемент – це елемент без текстового вмісту, без дочірнього елементу і без атрибута. Зазвичай відсутність елементу в схемі XML не має жодного конкретного значення; це може означати, що інформація невідома, або її не визначено, або ж вона відсутня з іншої причини.
60. Хоча у стандарті BOIB [ST.36](#) використовуються порожні елементи як своєрідні показники типу `Boolean` для позначення істинності або хибності умови залежно від наявності або відсутності елементу, стандарт BOIB ST.96 не передбачає їх використання. Натомість, необхідно створювати елементи, які чітко вказуватимуть на істинність або хибність умови, спростовуючи будь-які сумніви або усуваючи неоднозначність як під час зчитування машиною, так і під час сприйняття людиною.

- [SD-50] Порожні елементи НЕ ПОВИННІ визначатися у схемах, окрім як для розривання рядків.

3.2.3 Атрибути

61. Атрибути є структурними компонентами схеми W3C, пов'язаними з елементами, що надають додаткову інформацію стосовно цих елементів. На відміну від елементів, атрибути не можуть містити інші атрибути, тобто підрядних атрибутів не існує: атрибути не можуть бути розширені на зразок елементів.
62. Кардинальність атрибутів відрізняється від кардинальності елементів. Для атрибута може бути визначений показник `use` з одним із таких значень: "required", "optional" або "prohibited".

- [SD-51] Атрибут `use` НЕ МАЄ використовуватись для атрибутів, якщо атрибут, на який посилаються, є обов'язковим, оскільки `use="optional"` є значенням за замовчуванням.

3.2.4 Групування елементів та атрибутів

63. Формувальники – це структурні компоненти схеми W3C, що угруповують оголошення елементів. Стандарт схеми W3C передбачає три типи формувальників, а саме, `sequence`, `choice` та `all`.
64. Формувальник `sequence` вказує на те, що елементи, оголошені в ньому, наводяться в екземплярі документа XML в заявленому порядку. Формувальник `sequence` передбачає примусову реалізацію порядку елементів.
65. Формувальник `choice` вказує на те, що лише один елемент, оголошений в ньому, може наводитися в екземплярі XML.
66. У деяких випадках, показники появи у формувальниках `sequence` і `choice` передбачають гнучкі можливості використання формувальника `sequence/choice`, наприклад, необов'язкове використання або багаторазове використання, для уникнення необхідності визначення додаткового рівня.
67. Формувальник `all` вказує на те, що елементи, оголошені в ньому, можуть наводитися в екземплярі XML лише один раз, але у будь-якому порядку. В межах формувальника `all`, жодний елемент не може наводитися більше одного разу і всі елементи є обов'язковими, якщо атрибут `minOccurs` має значення 0. З метою обміну даними, принаймні один елемент має бути обов'язковим. Таким чином, формувальник `all` не рекомендований до використання у схемах.

- [SD-52] Формувальник `all` НЕ МАЄ використовуватися у схемах.

- [SD-53] Показники появи МОЖУТЬ використовуватися у схемах у формувальниках `xsd:sequence` та `xsd:choice`.

3.3 Розширення та обмеження

68. Деякі дозволені рекомендаціями W3C методи можуть впливати на гармонізацію даних, оскільки вони додають гнучкості у постачанні структурованих даних. Такими методами є: розширення, обмеження і використання груп заміни.

3.3.1 Розширення

69. Розширення комплексних типів не впливає на гармонізацію даних на відміну від розширення простих типів, оскільки компоненти, що додаються, не підлягають обміну. Однак, отримання простих типів за допомогою `union` або `list` порушує взаємодію, тому що вони допускають появу нових значень, непередбачених форматом обміну даними.

[SD-54] Схеми НЕ ПОВИННІ містити комплексні типи з `xsd:any` для забезпечення можливості розширення.

3.3.2 Обмеження

70. З точки зору гармонізації даних, обмеження комплексних типів модифікує структуру і може призвести до проблем із сумісністю, що перешкоджатиме процесу обміну даними. У випадку простих типів, обмеження модифікує значення; під час обміну даними не має виникнути жодних проблем.

3.3.3 Групи заміщення

71. Групи заміщення надають можливість замінити один глобальний елемент в екземплярі XML іншим глобальним елементом без необхідності додаткового коригування схеми. Однак, групи заміщення не забезпечуються гармонізацію імен елементів. Гармонізація є основною умовою взаємодіючого обміну даними, а використання груп заміщення не сприяє цьому.

[SD-55] У схемах НЕ ПОВИННІ використовуватися групи заміщення.

3.4 Обмеження ідентифікації

72. Як і будь-яка система зберігання, документ XML потребує наявності способів ідентифікації та посилання на інформацію, що міститься в ньому. Для цього у схемі W3C передбачено дві можливості, а саме `xsd:key/unique/keyref` і `xsd:ID/IDREF/IDREFS`.

73. Для посилань у межах поточного документа XML, `xsd:ID/IDREF/IDREFS` МОЖЕ співіснувати з `xsd:key/unique/keyref` у тому ж самому екземплярі XML. `xsd:key/unique/keyref` МАЄ використовуватись для кожного компонента в області, якщо єдиність повинна повністю зберігатися в цій області. `xsd:ID`, `xsd:IDREF` та `xsd:IDREFS` МАЮТЬ використовуватись в екземплярах XML, якщо єдиність і посилання на єдиність необхідні для всього документа. Атрибути `id`, `idref` та `idrefs` визначаються у просторі імен *Common* за допомогою `xsd:ID`, `xsd:IDREF` та `xsd:IDREFS` відповідно. Атрибут `idrefs` МАЄ використовуватись, якщо очікується множина посилань, наприклад, у формулі.

74. Для посилань, які є зовнішніми для поточного документа XML, наприклад для перехресного посилання, МАЄ використовуватись атрибут `extRef`, що визначається як `xsd:token` у просторі імен *Common*.

- [SD-56] Схеми МАЮТЬ використовувати `xsd:key/xsd:keyref/xsd:unique` та/або `xsd:ID/xsd:IDREF/xsd:IDREFS` для ідентифікації обмежень, якщо необхідно.
- [SD-57] Схеми МАЮТЬ використовувати `xsd:ID/xsd:IDREF/xsd:IDREFS` для одного посилання або множини посилань у межах поточного документа XML для ідентифікації обмежень, якщо необхідно.
- [SD-62] Схеми МАЮТЬ використовувати `xsd:key/xsd:unique/xsd:keyref` в області, якщо єдиність повинна повністю зберігатися в цій області.

3.5 Документування схем

75. Специфікація схеми W3C дозволяє включати до схем документування. Це дозволяє схемам містити вбудовану документальну інформацію для кожного структурного компонента. Кожний визначений або оголошений структурний компонент схеми **ПОВИНЕН** використовувати елемент `xsd:documentation` для відповідних документів та, якщо необхідно, елемент `xsd:appInfo` для інформацію про заявку в межах елемента `xsd:annotation`. Елемент `xsd:appInfo` містить роз'яснення про зміст і спосіб використання елементів у схемах для користувачів, які не володіють такими ж знаннями або не мають такої ж кваліфікації, що і розробник схем. Загалом, елемент `xsd:documentation` використовується для матеріалів, що сприймаються людиною, а елемент `xsd:appInfo` використовується для представлення інформації про інструменти, таблиці стилів та інші пристосовання.

3.5.1. Документування структурних компонентів схем

76. Схеми XML можуть також містити коментарі (тобто, `<!-- comment -->`). Розробникам рекомендується уникати їх використання, особливо, якщо мета коментарів полягає в описі особливостей певних структурних компонентів схеми.

77. Іноді допускається використання коментарів у форматі HTML, коли розробник бажає візуально розділити структурні компоненти, такі як елементи і типи.

- [SD-58] Усі схеми МАЮТЬ включати документування структурних компонентів схеми, використовуючи елемент `xsd:documentation`. Документування МАЄ лише описувати елементи або типи і НЕ МАЄ стосуватися подробиць їх реалізації або інших питань, безпосередньо не пов'язаних з тим, що означає той чи інший структурний компонент.
- [SD-59] Схема НЕ МАЄ містити коментарі (наприклад, `<!-- comment -->`).
- [SD-60] Документування НЕ МАЄ замінюватися на список кодів, які використовують переліки.

3.5.2 Документування заголовків схем

78. Так само як документація структурних компонентів схеми додає схемі ясності, документування у заголовку стосується призначення, способів використання і змісту схеми, і розміщується в одному місці у межах схеми. Документування заголовка схеми має бути включене до всіх елементів документального рівня і до деяких компонентів сукупного і базового рівнів, таких як *WIPOST3Code*. З метою спрощення обробки інформації у заголовку пристосованнями, для документування заголовка схеми має використовуватись елемент `xsd:appInfo`.

79. Таблиця 1 містить елементи, які включаються до заголовків усіх схем.

Таблиця 1. Інформація про документування заголовка

Елемент заголовка	Опис
Created date	Дата створення схеми (обов'язковий)
Last modified date	Дата останньої модифікації для схеми (обов'язковий)
Contact Point	Контактна особа для вирішення питань стосовно схеми (обов'язковий)
Release Notes URL	URL-адреса, за якою опубліковано інформацію стосовно версії схеми (необов'язковий)

У наступному прикладі показано елементи схеми, що відповідають елементам заголовка, наведеним у таблиці вище:

```
<xsd:annotation>
<xsd:appinfo> <com:SchemaCreatedDate>2011-05-28</com: SchemaCreatedDate>
<com:SchemaLastModifiedDate>2011-05-28</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>
```

80. Документування у заголовку схеми дозволяє розробнику легко визначити призначення, спосіб використання і зміст схеми. Ця інформація є також доволі корисною, коли розробникові потрібно вибрати схему для використання як шаблону під час створення іншої схеми.

[SD-61] Схеми документального рівня ПОВИННІ включати документування заголовків з посиланнями на дату створення, дату останньої модифікації та контактну особу, як обов'язкові елементи, а також, необов'язково, на URL-адресу, за якою опубліковано інформацію стосовно версій схем.

4. ПРАВИЛА ПРОЕКТУВАННЯ ЕКЗЕМПЛЯРІВ XML

81. Правила проектування екземплярів XML містять рекомендації, яких слід дотримуватися під час формування структурованих даних.

4.1 Простори імен в екземплярах XML

4.1.1 Валідація екземплярів XML

82. Екземпляри XML рекомендовано перевіряти на відповідність схемі. Валідація екземпляра XML забезпечує відповідність його змісту всім вимогам схеми, що використовується для перевірки. Система валідації вибирає схему, зважаючи на її точне місцезнаходження, вказане в екземплярі XML.

83. Місцезнаходження схеми повинне зазначатися у формі URI². Місцезнаходження схеми зазвичай вказується як URL³ на основі URI через обмеження можливості визначення URN⁴ на основі URI.

84. Місцезнаходження схеми може вказуватися в корінному елементі екземпляра XML, наприклад:

```
<?xml version="1.0" encoding="UTF-8"?>
<com:Contact xmlns:com="http://www.wipo.int/standards/XMLSchema/ST96/Common"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://www.wipo.int/standards/XMLSchema/ST96/Common
../V2_0/Common/Contact.xsd">
.....
</com:Contact>
```

85. Атрибут `schemaLocation` – це структурний компонент схеми W3C, що пов'язує екземпляр документа XML зі схемою. Він використовується, лише якщо схема має заданий простір імен. У прикладі, наведеному вище, ідентифікатор простору імен "<http://www.wipo.int/standards/XMLSchema/ST96/Common/1>" є заданим простором імен схеми "Contact.xsd".

- [ID-01] Екземпляри XML ПОВИННИ бути перевірені на відповідність певній схемі.
- [ID-02] В екземплярах документа XML ПОВИНЕН використовуватися атрибут `schemaLocation` для зв'язування цільового простору імен і місцезнаходження файлу схеми, якій відповідає екземпляр XML.
- [ID-03] Кожне оголошення атрибута `xsi:schemaLocation` МАЄ містити URL-адресу, за якою можна визначити ресурс.

4.1.2 Оголошення просторів імен і кваліфікація в екземплярах документів XML

² URI – *Uniform Resource Identifier* (уніфікований ідентифікатор ресурсу).

³ URL – *Uniform Resource Locator* (уніфікований визначник місцезнаходження ресурсу).

⁴ URN – *Uniform Resource Name* (уніфіковане ім'я ресурсу).

86. Так само як і у схемах, простір імен в екземплярах документів XML оголошується в його корінному елементі за допомогою ідентифікатора простору імен разом з рекомендованим префіксом простору імен. В екземплярах документів XML кваліфікація повинна використовуватись кваліфікація просторів імен для всіх елементів, визначених у Додатку III до стандарту BOIB ST.96.

87. Необхідно зазначити, що

- ідентифікатор простору імен в екземплярі документа XML повинен співпадати з ідентифікатором простору імен для заданого простору імен у схемі;
- префікс простору імен в екземплярі документа XML не повинен обов'язково співпадати з префіксом простору імен для заданого простору імен у схемі.

88. Елементи та атрибути в екземплярах документів XML можуть бути кваліфіковані простором імен, лише якщо вони належать до заданого простору імен схеми, на відповідність якій перевіряється той чи інший екземпляр документа XML. Тому, усі глобальні елементи та атрибути повинні бути кваліфіковані простором імен. Проте, ця вимога, якщо вона стосується локальних елементів та атрибутів, що належать до заданого простору імен схеми, залежить від налаштування механізму перемикання у схемі, що використовує два показники:

- `elementFormDefault`,
- `attributeFormDefault`.

89. Показник `elementFormDefault` контролює кваліфікацію простору імен локальних елементів, а показник `attributeFormDefault` контролює кваліфікацію простору імен локальних атрибутів. Обидва показники зазначаються як атрибути корінного елементу схеми і кожен з них має значення `qualified` (кваліфікований) або `unqualified` (некваліфікований).

Наприклад,

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

90. Ці оголошення вимагають, аби всі локальні елементи та атрибути у заданому просторі імен схеми були кваліфіковані простором імен в екземплярі документа XML, за винятком, якщо значення атрибута `form` було змінено.

- [ID-04] Кожному оголошеному простору імен МАЄ бути призначений рекомендований префікс, тобто, префікс `com` для всіх структурних компонентів схеми загального компонента, префікс `pat` для всіх структурних компонентів схеми компонента *Patent*, префікс `tmk` для всіх структурних компонентів схеми компонента *Trademark*, префікс `dgn` для всіх структурних компонентів схеми компонента *Design*, префікс `tbl` для структурних компонентів табличної схеми XML *OASIS* і префікс `mathml` для структурних компонентів схеми *MathML*.
- [ID-05] В екземплярах документів XML НЕ ПОВИННІ застосовуватися простори імен, які використовуються за замовчуванням.

4.1.3 Простори імен екземплярів схеми W3C

91. Стандарт схеми W3C має власний простір імен, відомий як простір імен екземплярів схеми W3C, що містить усі структурні компоненти схеми W3C, які використовуються в екземплярах документа XML (`schemaLocation`, `noNamespaceSchemaLocation`, `type` та `nil`). Для використання цих структурних компонентів, необхідно оголосити простір імен схеми W3C в корінному елементі екземпляра документа XML, використовуючи ідентифікатор простору імен "<http://www.w3.org/2001/XMLSchema-instance>":

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

92. Хоча префікс `xsi` визначається користувачем, він найчастіше використовується у літературі та документації схеми W3C як префікс просторів імен для структурних компонентів екземпляра схеми W3C.

- [ID-06] Екземпляри документів XML МАЮТЬ використовувати префікс `xsi` як префікс просторів імен для всіх структурних компонентів схеми W3C.

4.1.4 Видимість просторів імен

93. Простори імен в екземплярах документів XML мають видимість охоплення щодо застосування у екземплярах документів XML. Видимість простору імен застосовується для оголошеного елемента (яким може бути корінний елемент) і весь вміст цього елемента. Простір імен може бути оголошений також і для інших елементів, а не лише для корінного; це називається оголошенням локального простору імен.

94. Хоча оголошення локального простору імен може сприяти ефективності обробки, проте можливості візуальної ідентифікації всіх просторів імен, оголошених в екземплярі документа шляхом перевірки корінного елемента є важливішою.

- [ID-07] В екземплярах документів XML НЕ ПОВИННІ використовуватися оголошення локальних просторів імен.

4.2 Зовнішні сутності

95. Вбудоване зображення – це найпоширеніша зовнішня сутність, тобто посилання на зовнішній файл зображення, що вставляється в екземпляр документа XML там, де має відображатися зображення під час відтворення екземпляра. Вбудовані зображення є найбільш розповсюдженими частинами документів, які не можуть бути закодовані або збережені, використовуючи набір символів. Вбудованими зображеннями МОЖУТЬ бути креслення, хімічні формули, складні таблиці, невизначені символи тощо.

96. Екземпляр XML може містити посилання на інші файли XML на основі DTD, як зовнішню сутність. У такому випадку, екземпляр XML не може бути перевірений на відповідність певній схемі XML. Для посилання на файл XML може використовуватися `xsd:anyURI` або `xsd:string`, що надає можливість вказати місцезнаходження зовнішнього файла екземпляра XML. Проте, в екземплярах XML не рекомендується наводити посилання на файли XML на основі DTD.

97. Зображення можуть вбудовуватися в екземпляр XML як зображення, закодовані за допомогою вбудованого типу даних схеми W3C `base64Binary`, а також як посилання на зовнішні файли зображень, тобто зовнішні сутності. Однак, згідно з цим документом, зображення мають бути зовнішніми сутностями, оскільки бінарні зображення можуть містити небезпечний код, наприклад віруси.

- [ID-08] Зовнішні файли зображень МАЮТЬ бути в одному з форматів, рекомендованих стандартом VOIB ST.96.
- [ID-09] Зображення ПОВИННІ зберігатися як окремі файли.
- [ID-10] Переліки послідовностей МАЮТЬ відповідати рекомендаціям стандарту VOIB ST.25.

ДОПОВНЕННЯ А – СПИСОК ПРАВИЛ ПРОЕКТУВАННЯ

Загальні правила проектування

Ідентифікатор	Правило
[GD-01]	Усі схеми XML ПОВИННІ ґрунтуватися на технічних специфікаціях схеми W3C, що мають статус рекомендацій.
[GD-02]	Схеми ПОВИННІ відповідати вимогам таких документів: Схема XML. Частина 1. Структури (http://www.w3.org/TR/xmlschema-1) і Схема XML. Частина 2. Типи даних (http://www.w3.org/TR/xmlschema-2).
[GD-03]	Для символів ПОВИНЕН використовуватися стандарт ISO/IEC 10646 – Універсальний набір символів – <i>Unicode</i> . Символи <i>Unicode</i> ПОВИННІ кодуватися в UTF-8.
[GD-04]	Імена типів, елементів та атрибутів ПОВИННІ наводитися англійськими словами і відповідати вимогам до написання, зазначеним в Оксфордському словнику англійської мови (<i>Oxford English Dictionary</i>). Допустимими винятками є акроніми, аббревіатури та інші скорочення слів, наведені у Доповненні С.
[GD-05]	Імена типів, елементів та атрибутів МАЮТЬ складатися лише з іменників, прикметників і дієслів у теперішньому часі, за винятком акронімів, аббревіатур та інших скорочень слів, наведених у Доповненні С.
[GD-06]	Символи для використання в іменах типів, елементів або атрибутів ПОВИННІ знаходитися у межах такого набору: 'a-z, A-Z і 0-9'.
[GD-07]	Максимальна довжина імені компонента МАЄ складати не більше 35 символів.
[GD-08]	Імена типів, елементів та атрибутів МАЮТЬ бути стислими та очевидними.
[GD-09]	В іменах елементів ПОВИННА використовуватись угода upper camel case (UCC). Наприклад, <i>CountryCode</i> .
[GD-10]	В іменах типів ПОВИННА використовуватись угода UCC + суфікс "Type". Наприклад, <i>ApplicantType</i> .
[GD-11]	В іменах атрибутів ПОВИННА використовуватись угода lower camel case (LCC). Наприклад, <i>currencyCode="EUR"</i> .
[GD-12]	Акроніми та аббревіатури, наведені у Доповненні С, ПОВИННІ завжди використовуватися замість повних розширених імен.
[GD-13]	Акроніми та аббревіатури на початку оголошення атрибута ПОВИННІ наводитися у нижньому регістрі. Всі інші акроніми та аббревіатури в

	оголошенні атрибута ПОВИННІ наводитися у верхньому реєстрі як зазначено у Доповненні С.
[GD-14]	Акроніми та аббревіатури для імен елементів і типів ПОВИННІ наводитися відповідно до Доповнення С.
[GD-15]	Імена комплексних типів МАЮТЬ містити значущі об'єктні класи.
[GD-16]	Якщо, зважаючи на вид суб'єкта господарювання, ім'я певного компонента має містити два терміни об'єктних класів, можна використовувати комплексний поєднаний тип. Ім'я комплексного поєданого типу МАЄ складатися з об'єктного класу комплексного поєднуючого типу, властивості, що описує характер поєднання, та об'єктного класу комплексного поєднуваного типу. Кваліфікатори можуть передувати об'єктному класу і властивості. Наприклад, в імені <code>ApplicantResidenceAddress</code> : <code>Applicant</code> є об'єктним класом комплексного поєднуючого типу, <code>Residence</code> є властивістю, а <code>Address</code> – об'єктним класом комплексного поєднуваного типу.
[GD-17]	Імена простих елементів (що базуються на простому або комплексному типі, який не допускає дочірніх елементів) МАЮТЬ складатися з об'єктного класу, властивості та, за наявності, терміну представлення. Кваліфікатори можуть передувати об'єктному класу та властивості.
[GD-18]	Об'єктний клас завжди ПОВИНЕН мати однакове семантичне значення у межах всього простору імен і МОЖЕ складатися з декількох слів. Наприклад, <code>ContactInformation</code> .
[GD-19]	Властивість в імені ПОВИННА бути унікальною в межах об'єктного класу, проте МОЖЕ використовуватися повторно в різних об'єктних класах.
[GD-20]	Кваліфікатор МОЖЕ, за необхідності, додаватися до об'єктного класу або властивості для індивідуалізації імені.
[GD-21]	Якщо ім'я містить об'єктний клас, властивість і термін представлення, об'єктний клас ПОВИНЕН передувати властивості, а властивість ПОВИННА передувати терміну представлення. Кваліфікатор МАЄ передувати поєднуваному об'єктному класу та властивості.
[GD-22]	Якщо властивість закінчується тим самим (або ж еквівалентним) словом, що і термін представлення, то термін представлення ПОВИНЕН бути видалений.
[GD-23]	Якщо потрібен термін представлення, терміни представлення, наведені у Доповненні В, ПОВИННІ використовуватися як терміни представлення в іменах базових компонентів.
[GD-24]	Усі імена типів, елементів та атрибутів у просторі імен ПОВИННІ бути унікальними.

[GD-25]	Слово або слова в імені МАЮТЬ наводитися в однині, за винятком, якщо поняття передається лише множиною. Наприклад, <i>TotalMarkSeries</i> .
[GD-26]	Ім'я елемента або типу, що містить сукупність контекстуально пов'язаних компонентів, МАЄ включати суфікс "Bag". Наприклад, <i>EmailAddressBag</i> означає сукупність елементів <i>EmailAddress</i> .
[GD-27]	Сполучні слова на зразок сполучника "and", прийменника "of" та артикля "the" НЕ МАЮТЬ використовуватися в іменах типів, елементів та атрибутів, за винятком, якщо вони складають частину виробничої термінології.
[GD-28]	Імена типів, елементів та атрибутів НЕ ПОВИННІ перекладатися, змінюватися або замінюватися за жодних умов.
[GD-29]	Імена типів та елементів НЕ ПОВИННІ містити посилання на номери статей і правил. Наприклад, <i>PCTRule702C</i> .
[GD-30]	Символи для використання в іменах файлів схем ПОВИННІ знаходитися у межах такого набору: 'a-z, A-Z, 0-9, знак підкреслювання "_" і крапка "."'.
[GD-31]	Назва файла схеми ПОВИННА складатися з двох обов'язкових частин з одним роздільником та необов'язкової інформації про її версію з двома додатковими роздільниками, тобто: <назва компонента>{"_" "V"<номер основної версії>"_"<номер проміжної версії>"}."<розширення файла>. Наприклад, <i>EmailAddressType.xsd</i> , <i>languageCode.xsd</i> , <i>ApplicationBody_V1_0.xsd</i> .
[GD-32]	Назва файла проекту схеми ПОВИННА складатися з чотирьох обов'язкових частин з двома роздільниками та необов'язкової інформації про її версію з двома додатковими роздільниками, тобто: <назва компонента>{"_" "V"<номер основної версії>"_"<номер проміжної версії>"}{"_" "D"<номер редакції>"}."<розширення файла>, наприклад, <i>Contact_D3.xsd</i> , <i>TrademarkApplication_V1_1_D1</i> . Якщо проект схеми базується на існуючій схемі, а назва його файла містить інформацію про версію схеми, номери основної та проміжної версій у назві файла проекту схеми МАЮТЬ відповідати номерам, зазначеним у назві файла схеми, на якій базується проект схеми. Якщо проект схеми є новим, номер основної версії у назві файла проекту схеми МАЄ відповідати номеру, визначеному в відповідному просторі імен, а номером проміжної версії у назві файла проекту схеми МАЄ бути "0".

Правила проектування схем

Ідентифікатор	Правило
[SD-01]	Модулі схем компонента типу <i>Patent</i> , модулі схем компонента типу

	<i>Trademark</i> і модулі схем компонента типу <i>Design</i> ПОВИННІ використовувати структурний компонент <code>xsd:import</code> для посилання на модулі схем компонента типу <i>Common</i> .
[SD-02]	Модулі схем компонента типу <i>Common</i> НЕ ПОВИННІ посилатися на модулі схем компонента типу <i>Patent</i> , модулі схем компонента типу <i>Trademark</i> або модулі схем компонента типу <i>Design</i> .
[SD-03]	Модуль схеми компонента типу <i>Patent</i> , <i>Trademark</i> або <i>Design</i> МОЖЕ посилатися на модулі схем компонента типу <i>Common</i> , на схеми у тому ж модулі компонента типу <i>Common</i> і на погоджені схеми за промисловими стандартами, проте він НЕ ПОВИНЕН посилатися на інші модулі схем компонента типу <i>Common</i> . Наприклад, схеми компонента типу <i>Patent</i> НЕ ПОВИННІ посилатися на схеми компонента типу <i>Trademark</i> і навпаки.
[SD-04]	Якщо можливо, ПОВИННІ використовуватися існуючі схеми, а не створюватися нові.
[SD-05]	Схеми МАЮТЬ максимально використовувати елементи і типи, визначені в існуючих схемах.
[SD-06]	Усі типи, елементи та атрибути ПОВИННІ оголошуватися глобально.
[SD-07]	Схеми НЕ ПОВИННІ використовувати структурний компонент <code>xsd:redefine</code> .
[SD-08]	Схеми ПОВИННІ використовувати простори імен.
[SD-09]	Опубліковані оголошення просторів імен НЕ МАЮТЬ змінюватися.
[SD-10]	Схеми ПОВИННІ містити оголошення простору імен схеми W3C.
[SD-11]	Схеми ПОВИННІ використовувати кваліфікації просторів імен для всіх структурних компонентів схеми W3C.
[SD-12]	Схеми ПОВИННІ використовувати простір імен у форматі URI.
[SD-13]	Усі схеми ПОВИННІ мати атрибути <code>elementFormDefault</code> і <code>attributeFormDefault</code> зі значенням "qualified" у кореневому елементі <code>xsd:schema</code> .
[SD-14]	Назви просторів імен ПОВИННІ мати таку структуру: <a href="http://www.wipo.int/standards/XMLSchema/<category>">http://www.wipo.int/standards/XMLSchema/<category> , де <code><category></code> – це ідентифікатор використовуваного домену схеми: <i>Common</i> , <i>Patent</i> , <i>Trademark</i> або <i>Design</i> .
[SD-15]	Схеми МАЮТЬ використовувати "xsd" як префікс простору імен для всіх структурних компонентів схеми W3C, "com" для всіх схем компонента типу <i>Common</i> , "pat" для всіх схем компонента типу <i>Patent</i> , "tmk" для всіх схем компонента типу <i>Trademark</i> , "dgn" для всіх схем компонента типу <i>Design</i> ,

	"tbl" для схем табличного компонента <i>OASIS</i> і "mathml" для схем <i>MathML</i> .
[SD-16]	Кожна схема ПОВИННА містити оголошення цільового простору імен з використанням атрибута <code>xsd:targetNamespace</code> .
[SD-17]	Атрибут схеми <code>targetNamespace</code> ПОВИНЕН відповідати назві одного з оголошених просторів імен, окрім простору імен схеми W3C.
[SD-18]	Схеми компонента типу <i>Common</i> ПОВИННІ використовувати http://www.wipo.int/standards/XMLSchema/Common як цільовий простір імен.
[SD-19]	Схеми компонента типу <i>Patent</i> ПОВИННІ використовувати http://www.wipo.int/standards/XMLSchema/Patent як цільовий простір імен.
[SD-20]	Схеми компонента типу <i>Trademark</i> ПОВИННІ використовувати http://www.wipo.int/standards/XMLSchema/Trademark як цільовий простір імен.
[SD-21]	Схеми компонента типу <i>Design</i> ПОВИННІ використовувати http://www.wipo.int/standards/XMLSchema/Design як цільовий простір імен.
[SD-22]	Схеми НЕ МАЮТЬ використовувати простір імен за замовчуванням.
[SD-23]	Нові номери проміжних версій схем ПОВИННІ забезпечувати відповідність усіх екземплярів, створених на підставі попередніх проміжних версій схем, одній основній версії.
[SD-24]	Якщо нова схема не забезпечує відповідність існуючого екземпляра XML, створеного на підставі поточної основної версії, номер основної версії схеми ПОВИНЕН бути збільшений.
[SD-25]	<i>Видалено</i>
[SD-26]	<i>Видалено</i>
[SD-27]	У випадку оновлення включеної або імпортованої схеми, номер основної та/або проміжної версії схеми МАЄ бути змінений.
[SD-28]	Якщо будь-який структурний компонент схеми зазнав змін, номери версій усіх схем у випуску ПОВИННІ бути збільшені.
[SD-29]	Під час створення нової схеми МАЮТЬ використовуватися включені або імпортовані схеми останніх версій.
[SD-30]	Атрибут <code>version</code> схеми W3C ПОВИНЕН складатися з номера основної версії та номера проміжної версії для кожного файлу схеми у такому форматі: "V"<номер основної версії>"_"<номер проміжної версії>.
[SD-31]	Схема для компонента Document ПОВИННА визначати обов'язковий

	атрибут з іменем <code>st96Version</code> у своєму кореневому елементі з постійним значенням, що відповідає версії випуску схеми XML за стандартом BOIB ST.96 у такому форматі: "V"<номер основної версії>"_"<номер проміжної версії>.
[SD-32]	Схеми за стандартом BOIB ST.96 МАЮТЬ розроблюватися з однаковим рівнем кардинальності ⁵ та деталізації для спрощення взаємоперетворення екземплярів XML за стандартом BOIB ST.96 на екземпляри XML за стандартом BOIB ST.36 , ST.66 або ST.86 .
[SD-33]	Переліки значень у схемах НЕ МАЮТЬ включати значення "Other" або "Unknown", якщо відсутня необхідність.
[SD-34]	Для математичних формул МАЄ використовуватись схема <i>MathML</i> версії 3.
[SD-35]	Для таблиць МАЄ використовуватись схема <i>OASISTable.xsd</i> , створена на основі схеми <i>OASIS Exchange Table</i> .
[SD-36]	Схеми промислових стандартів МАЮТЬ включатися лише за допомогою посилання і після попереднього погодження Комітету зі стандартів BOIB.
[SD-37]	Схеми МАЮТЬ максимально використовувати прості типи.
[SD-38]	Списки кодів МАЮТЬ оголошуватися як переліки значень у простому типі.
[SD-39]	Список кодів МОЖЕ бути оголошений як поєднання простих типів.
[SD-40]	Для представлення ВПВ, пріоритетних і зазначених країн/організацій ПОВИННІ використовуватися двобуквені коди за стандартом BOIB ST.3 . Наприклад, <code>PriorityCountryCode="EP"</code> .
[SD-41]	Для представлення назв країн, колоній та інших територій, що представляють певний геополітичний інтерес, ПОВИННІ використовуватися кодувальні елементи Альфа-2 за стандартом ISO 3166-1 (двобуквені коди назв країн) на основі списків назв країн, отриманих від Організації Об'єднаних Націй.
[SD-42]	Для кодування мов ПОВИННІ використовуватися коди за стандартом ISO 639-1 (двобуквені коди мов).
[SD-43]	Схеми ПОВИННІ оголошувати елементи та атрибути для дати і часу з використанням типів даних часу і дати за схемою W3C.
[SD-44]	Для кодування валют ПОВИННІ використовуватися коди за стандартом ISO 4217 (трибуквені коди валют).
[SD-45]	Символи, що використовуються у переліках значень, ПОВИННІ обмежуватися таким набором: 'a-z, A-Z, 0-9, пробіл " " і знак

⁵ Тут "кардинальність" означає кількість елементів у наборі.

	підкреслювання "_". Перелічувані значення НЕ МАЮТЬ починатися з цифрового символу.
[SD-46]	Перелічувані значення МАЮТЬ бути "семантично достатніми" і наводитися англійською мовою якомога меншою кількістю символів. Значення МАЮТЬ походити із загальноприйнятої термінології у сфері промислової власності.
[SD-47]	МОЖУТЬ використовуватися абстрактні комплексні типи.
[SD-48]	Схеми МАЮТЬ використовувати атрибути лише для визначення невиробничих даних. Наприклад, для номерів послідовностей допустимо використовувати атрибут <code>sequenceNumber</code> .
[SD-49]	Показник появи НЕ МАЄ використовуватись для зазначення обмеження, яке є стандартним у межах схеми. Наприклад, <code>minOccurs="1"</code> і <code>maxOccurs="1"</code> НЕ МАЮТЬ використовуватись, проте <code>minOccurs="2"</code> і <code>maxOccurs="3"</code> є прийнятними.
[SD-50]	Порожні елементи НЕ ПОВИННІ визначатися у схемах, окрім як для розривання рядків.
[SD-51]	Атрибут <code>use</code> НЕ МАЄ використовуватись для атрибутів, якщо атрибут, на який посилаються, є необов'язковим, оскільки <code>use="optional"</code> є значенням за замовчуванням.
[SD-52]	Формувальник <code>all</code> НЕ МАЄ використовуватися у схемах.
[SD-53]	Показники появи МОЖУТЬ використовуватися у схемах у формувальниках <code>xsd:sequence</code> та <code>xsd:choice</code> .
[SD-54]	Схеми НЕ ПОВИННІ містити комплексні типи з <code>xsd:any</code> для забезпечення можливості розширення.
[SD-55]	У схемах НЕ ПОВИННІ використовуватися групи заміщення.
[SD-56]	Схеми МАЮТЬ використовувати <code>xsd:key/xsd:keyref/xsd:unique</code> та/або <code>xsd:ID/xsd:IDREF/xsd:IDREFS</code> для ідентифікації обмежень, якщо необхідно.
[SD-57]	Схеми МАЮТЬ використовувати <code>xsd:ID/xsd:IDREF/xsd:IDREFS</code> для одного посилання або множини посилань у межах поточного документа XML для ідентифікації обмежень, якщо необхідно.
[SD-58]	Усі схеми МАЮТЬ включати документування структурних компонентів схеми, використовуючи елемент <code>xsd:documentation</code> . Документування МАЄ лише описувати елементи або типи і НЕ МАЄ стосуватися подробиць їх реалізації або інших питань, безпосередньо не пов'язаних з тим, що означає той чи інший структурний компонент.
[SD-59]	Схема НЕ МАЄ містити коментарі (наприклад, <code><!-- comment --></code>).

[SD-60]	Документування НЕ МАЄ замінюватися на список кодів, які використовують переліки.
[SD-61]	Схеми документального рівня ПОВИННІ включати документування заголовків з посиланнями на дату створення, дату останньої модифікації та контактну особу, як обов'язкові елементи, а також, необов'язково, на URL-адресу, за якою опубліковано інформацію стосовно версій схем.
[SD-62]	Схеми МАЮТЬ використовувати <code>xsd:key/xsd:unique/xsd:keyref</code> в області, якщо єдиність повинна повністю зберігатися в цій області.
[SD-63]	Схема для компонента <i>Document</i> ПОВИННА оголошувати необов'язковий атрибут <code>ipoVersion</code> у своєму кореневому елементі з постійним значенням, що відповідає версії випуску відомчої схеми у такому форматі: <code><ST.3 Code>"_""V"<номер основної версії>"_ "<номер проміжної версії></code> , наприклад <code>"US_V2_0"</code> .
[SD-64]	Папка випуску схеми, ім'я файла схеми на рівні документа та ім'я файла зведеної схеми ПОВИННІ містити відповідну інформацію про версію, що складається з номера основної версії та номера проміжної версії.

Правила проектування екземплярів

Ідентифікатор	Правило
[ID-01]	Екземпляри XML ПОВИННІ бути перевірені на відповідність певній схемі.
[ID-02]	В екземплярах документа XML ПОВИНЕН використовуватися атрибут <code>schemaLocation</code> для зв'язування цільового простору імен і місцезнаходження файла схеми, якій відповідає екземпляр XML.
[ID-03]	Кожне оголошення атрибута <code>xsi:schemaLocation</code> МАЄ містити URL-адресу, за якою можна визначити ресурс.
[ID-04]	Кожному оголошеному простору імен МАЄ бути призначений рекомендований префікс, тобто, префікс <code>com</code> для всіх структурних компонентів схеми загального компонента, префікс <code>pat</code> для всіх структурних компонентів схеми компонента <i>Patent</i> , префікс <code>tmk</code> для всіх структурних компонентів схеми компонента <i>Trademark</i> , префікс <code>dgn</code> для всіх структурних компонентів схеми компонента <i>Design</i> , префікс <code>tbl</code> для структурних компонентів табличної схеми XML <i>OASIS</i> і префікс <code>mathml</code> для структурних компонентів схеми <i>MathML</i> .
[ID-05]	В екземплярах документів XML НЕ ПОВИННІ застосовуватися простори імен, які використовуються за замовчуванням.
[ID-06]	Екземпляри документів XML МАЮТЬ використовувати префікс <code>xsi</code> як

	префікс просторів імен для всіх структурних компонентів схеми W3C.
[ID-07]	В екземплярах документів XML НЕ ПОВИННІ використовуватися оголошення локальних просторів імен.
[ID-08]	Зовнішні файли зображень МАЮТЬ бути в одному з форматів, рекомендованих стандартом BOIB ST.96.
[ID-09]	Зображення ПОВИННІ зберігатися як окремі файли.
[ID-10]	Переліки послідовностей МАЮТЬ відповідати рекомендаціям стандарту BOIB ST.25 .

ДОПОВНЕННЯ В – ТЕРМІНИ ПРЕДСТАВЛЕННЯ

Термін	Визначення	Тип даних
Сума	Грошове значення.	AmountType
Категорія	Окремо визначена група або підмножина в системі класифікації, в якій усі складові підлягають одним і тим самим принципам таксономії.	xsd:token
Код	Комбінація однієї чи більше цифр, літер або спеціальних символів, яка замінюється певним значенням. Представляє собою обмежені, попередньо визначені значення або довільний формат.	xsd:token
Дата	Позначення певного моменту у часі, представлене роком, місяцем та днем.	DateType або xsd:date
Ідентифікатор	Комбінація одного або більше цілих чисел, літер або спеціальних символів, що ідентифікують певний примірник бізнес-об'єкта, але яка не має значення, що легко піддається визначенню.	xsd:token
Показник	Позначення присутності, відсутності або вимоги чого-небудь. Рекомендованими значеннями є "Y", "N" або, якщо треба, "?".	xsd:boolean
Міра	Числове значення, що визначається шляхом вимірювання об'єкта за допомогою певної одиниці вимірювання. MeasureType використовується для вираження фізичних параметрів об'єкта, таких як температура, довжина, швидкість, ширина, вага, об'єм, широта тощо. Іншими словами, MeasureType має використовуватися для вимірювання суттєвих або фізичних властивостей об'єкта в цілому.	MeasureType
Назва	Позначення об'єкта, виражене за допомогою слова або виразу.	xsd:string
Номер	Рядок цифрових або буквено-цифрових символів, за допомогою яких виражається мітка, значення, кількість або ідентифікація.	xsd:positiveInteger, xsd:string або xsd:token
Відсоток	Число, за допомогою якого виражається частина цілого, поділена на 100.	xsd:decimal

Кількість	Кількість є зліченим числом негрошових одиниць, включаючи, можливо, дробі. <code>Quantity</code> використовується для позначення зліченого числа предметів. <code>Quantity</code> має використовуватися для простих властивостей об'єкта, що тлумачиться як комбінація, зібрання або контейнер для визначення кількості або підрахування його складових. <code>Quantity</code> має завжди представляти собою злічене число предметів, а властивість може позначатися як "всього", "відправлено", "завантажено", "зберігається". <code>QuantityType</code> має використовуватись для компонентів, які потребують інформації про одиниці; а <code>xsd:nonNegativeInteger</code> має використовуватись для злічених компонентів, які не потребують інформації про одиниці.	<code>QuantityType</code> або <code>xsd:nonNegativeInteger</code>
Пропорція	Кількість або сума відносно іншої кількості або суми.	<code>xsd:decimal</code>
Текст	Неформатований рядок символів, зазвичай слів (включає аббревіатури і коментарі).	<code>xsd:string</code> , <code>LocalizedTextType</code> або <code>OrderedTextType</code>
Час	Позначення певного хронологічного моменту в певний період.	<code>xsd:time</code>
Дата і час	Зареєстрована дата і час, коли сталася подія.	<code>xsd:datetime</code>
URI	Уніфікований ідентифікатор ресурсів, що ідентифікує місцезнаходження файлу.	<code>xsd:anyURI</code>

ДОПОВНЕННЯ С – ПЕРЕЛІК АКРОНІМІВ ТА АБРЕВІАТУР

Загалом, що стосується аббревіатур, перша літера у слові або у вислові має бути великою, а інші – малими (наприклад, "Pre", "BioDeposit"), за винятком загальновідомих аббревіатур на кшталт "IDREF", "W3C", "ST3". У випадку акронімів усі літери мають бути великими, наприклад, "WIPO".

Alt	Альтернативний текст для зображення
B	Грубий шрифт
BioDeposit	Біологічне депонування
Br	Розрив рядка
CDX	Формат файлів програмного забезпечення <i>ChemDraw</i> (власність <i>CambridgeSoft</i>)
CPC	Спільна патентна класифікація
DD	Опис визначення
Del	Видалений текст
DL	Список визначень
DOI	Цифровий ідентифікатор об'єкта
DT	Визначення
DTD	Визначення типу документа
DWF	Формат файлів <i>Design Web Format</i>
DWG	Креслення
ECLA	Європейська класифікація
ExtRef	Посилання, що є зовнішніми для поточного документа XML
H<n>	"n" вказує на рівень заголовка з певним значенням від 1 до 15. Це означає, що ця аббревіатура у перелічуваному значенні представляє собою один з елементів у діапазоні від H1 до H15. Наприклад, "H1" означає "Заголовок 1".
I	Курсивний шрифт
ID	Ідентифікатор для системної ідентифікації
IDREF	Одне посилання на ідентифікатор
IDREFS	Множина посилань на ідентифікатори

IGES	Формат файлів <i>Initial Graphic Exchange Specification</i>
Ins	Вставлений текст
IP	Промислова власність
IPC	Міжнародна патентна класифікація
IPCR	Реформа Міжнародної патентної класифікації
IPO	Відомство промислової власності
IPR	Об'єкт промислової власності
ISO	Міжнародна організація зі стандартизації
LCC	Угода <i>Lower Camel Case</i>
LI	Елемент списку
MPEG	Експертна група з питань рухомих зображень
MOL	Формат файлів для зберігання інформації про атоми, зв'язки, поєднуваність і координати молекул
NB	Формат файлів записників програмного забезпечення <i>Mathematica</i>
NPL	Непатентна література
O	Закреслений текст
OASIS	Організація з розвитку стандартів у сфері структурованої інформації
OCR	Оптичне розпізнавання символів
OL	Упорядкований список
P	Параграф
PAN	Номер платіжної картки
PCT	Договір про патентну кооперацію
PKCS7	У криптографії , PKCS є групою стандартів щодо асиметричних алгоритмів шифрування , а PKCS #7 (PKCS7) означає стандарт щодо синтаксису криптографічних повідомлень, в якому описаний загальний синтаксис даних, до яких може застосовуватися криптографічний захист, наприклад цифрові підписи і цифрові конверти
Pre	Попередньо відформатований текст

S	Закреслений текст
ST3	Стандарт BOIB ST.3
ST13	Стандарт BOIB ST.13
Sub	Підрядковий текст
Sup	Надрядковий текст
SVG	Формат файлів <i>Scalable Vector Graphic</i>
SWF	Формат файлів <i>Small Web Format</i>
SWIFT	Товариство всесвітніх міжбанківських фінансових телекомунікацій
ThreeDM	Просторове моделювання
ThreeDS	Програмне забезпечення <i>3D Studio</i>
U	Підкреслений шрифт
UCC	Угода <i>Upper Camel Case</i>
UL	Невпорядкований список
UPOV	Міжнародний союз з охорони нових сортів рослин
URI	Уніфікований ідентифікатор ресурсу
URL	Уніфікований локатор ресурсу
URN	Уніфікована назва ресурсу
W3C	Консорціум Всесвітньої павутини
WIPO	Всесвітня організація інтелектуальної власності
WMV	Формат файлів <i>Windows Media Video</i>
XML	Розширювана мова розмітки

ДОПОВНЕННЯ D – ПОСИЛАННЯ

Стандарти BOIB

- Стандарт BOIB [ST.3](#) "Рекомендований стандарт стосовно двобуквених кодів для представлення держав, інших адміністративних одиниць та міжурядових організацій";
- Стандарт BOIB [ST.13](#) "Рекомендації щодо нумерації заявок на об'єкти промислової власності (ОПВ)";
- Стандарт BOIB [ST.25](#) "Стандарт з представлення переліків послідовностей нуклеотидів та амінокислот у патентних заявках";
- Стандарт BOIB [ST.36](#) "Рекомендації щодо обробки інформації стосовно винаходів, корисних моделей з використанням XML (розширюваної мови розмітки)";
- Стандарт BOIB [ST.66](#) "Рекомендації щодо обробки інформації стосовно торговельних марок з використанням XML (розширюваної мови розмітки)";
- Стандарт BOIB [ST.67](#) "Рекомендації щодо електронної обробки зображувальних елементів торговельних марок";
- Стандарт BOIB [ST.86](#) "Рекомендації щодо обробки інформації стосовно промислових зразків з використанням XML (розширюваної мови розмітки)".

Промислові стандарти

- Рекомендації щодо мови визначення схеми (XSD) W3C – [Схема XML. Частина 1. Структури](#) і [Схема XML. Частина 2. Типи даних](#);
- Запит про надання коментарів [\(RFC\) 2119](#), опублікований Робочою групою інженерів Інтернет (*Internet Engineering Task Force*);
- Стандарт [ISO/IEC 10646](#) "Інформаційні технології. Універсальний набір кодованих символів (UCS)";
- Стандарт [ISO 11179](#) "Інформаційні технології. Реєстри метаданих. Частина 5. Принципи іменування та ідентифікації";
- Стандарт [ISO 3166-1:2006](#) "Коди для представлення назв країн та їх частин. Частина 1. Коди країн";
- Стандарт [ISO 639-1:2002](#) "Коди для представлення назв мов. Частина 1. Код Альфа-2";
- Стандарт [ISO 8601:2004](#) "Елементи даних і формати для обміну інформацією. Обмін інформацією. Представлення дат і часу";
- Стандарт [ISO 4217:2008](#) "Коди для представлення валют і фондів";
- Таблична схема XML OASIS:
<http://www.oasis-open.org/docbook/xmlschema/1.0b1/calstbl.xsd>;
- Схема MathLab: MathML, версія 3: <http://www.w3.org/TR/MathML3>.

[Кінець Додатка I]