

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

Редакція 1.0

*прийнята Комітетом зі стандартів BOIV (КСВ)
на його другій сесії 4 травня 2012 року*

Зміст

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	6
1.5.3	Ідентифікатори правил	6
2.	УМОВИ ПРОЕКТУВАННЯ З ВИКОРИСТАННЯМ XML	8
2.1	Загальні правила проектування з використанням XML.....	8
2.2	Угоди про іменування в XML.....	8
2.2.1	Угоди про іменування структурних компонентів схем	8
2.2.2	Угоди про іменування файлів схем.....	10
2.3	Принцип модульної побудови	11
2.3.1	Модулі схем	11
2.3.2	Зовнішні посилання у схемах	12
2.4	Можливість повторного використання.....	14
2.5	Простори імен	15
2.5.1	Оголошення і кваліфікація просторів імен	15
2.5.2	Простори імен у схемах XML	15
2.5.3	Цільові простори імен.....	16
2.5.4	Простори імен за замовчуванням	17
2.6	Керування версіями схем.....	18

2.6.1	Значні та незначні зміни	18
2.6.1.1	Основні версії	18
2.6.1.2	Проміжні версії.....	18
2.6.2	Стратегія керування версіями схем	19
2.6.2.1	Простори імен у межах керування версіями схем.....	19
2.6.2.2	Угоди про іменування файлів у межах керування версіями схем	20
2.6.2.3	Вбудований у схему XML атрибут <code>version</code> у межах керування версіями схем.....	20
2.6.2.4	Атрибут <code>schemaVersion</code> , що визначається користувачем, у межах керування версіями схем для екземплярів XML.....	20
2.7	Взаємоперетворення з іншими стандартами BOIV, що стосуються XML.....	21
2.8	Схеми промислових стандартів	22
3.	УМОВИ ПОБУДОВИ СХЕМ XML	23
3.1	Визначення типів	23
3.1.1	Прості типи.....	23
3.1.1.1	Вбудовані типи даних схеми W3C	23
3.1.1.2	Користувацькі типи даних	23
3.1.2	Комплексні типи.....	24
3.2	Елементи та атрибути	24
3.2.1	Елементи або атрибути.....	24
3.2.2	Елементи.....	25
3.2.2.1	Кардинальність елементів	25
3.2.2.2	Порожні елементи	25
3.2.3	Атрибути.....	25
3.2.4	Групування елементів та атрибутів.....	26
3.3	Розширення та обмеження	26
3.3.1	Розширення	26
3.3.2	Обмеження	27
3.3.3	Групи заміщення	27
3.4	Обмеження ідентифікації.....	27
3.5	Документування схем.....	27
3.5.1	Документування заголовків схем	28
4.	ПРАВИЛА ПРОЕКТУВАННЯ ЕКЗЕМПЛЯРІВ XML.....	30
4.1	Простори імен в екземплярах XML	30

4.1.1	Валідація екземплярів XML.....	30
4.1.2	Оголошення просторів імен і кваліфікація в екземплярах документів XML...	31
4.1.3	Простори імен екземплярів XML схеми W3C	32
4.1.4	Видимість просторів імен.....	32
4.2	Зовнішні сутності.....	33
ДОПОВНЕННЯ А – ЗВЕДЕНИЙ СПИСОК ПРАВИЛ ПРОЕКТУВАННЯ		34
	Загальні правила проектування	34
	Правила проектування схем.....	36
	Правила проектування екземплярів.....	40
ДОПОВНЕННЯ В – ТЕРМІНИ ПРЕДСТАВЛЕННЯ		42
ДОПОВНЕННЯ С – ПЕРЕЛІК АБРЕВІАТУР І СКОРОЧЕНЬ		44
ДОПОВНЕННЯ D – ПОСИЛАННЯ		46
	Стандарти VOIB.....	46
	Промислові стандарти	46

1. ВСТУП

1.1 Огляд

1. Цей стандарт призначений для забезпечення узгодженості трьох об'єктів промислової власності 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 умов проектування для створення і використання схем та екземплярів XML, застосовних до всіх видів 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 ширшою спільнотою, залученою до сфери промислової власності. Вони повинні використовувати цей документ як вихідні правила та умови проектування, яким мають відповідати усі схеми XML, аби вони вважалися сумісними зі стандартом BOIV 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 рекомендується 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 «Умови проектування з використанням XML» визначає загальні правила, що застосовуються до проектування як схем, так і екземплярів XML;
- Розділ 3 «Умови створення 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 аббревіатур і скорочень», що містить аббревіатури і скорочення, які використовуються в цьому документі;
- Доповнення D «Посилання», що містить посилання на стандарти BOIB 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*;
- вираз «схема, сумісна зі стандартом BOIB ST.96» стосується схеми, що відповідає компонентам схеми за стандартом BOIB ST.96 і Правилам та умовам проектування з використанням XML у сфері промислової власності, тобто вимогам Додатку I стандарту BOIB ST.96;
- вираз «схема, адаптована до стандарту BOIB ST.96» стосується сумісної схеми, яку не було розширено і яка містить обмеження, що відповідають обмеженням у схемі за стандартом BOIB ST.96.

7. У цьому документі

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

8. Ці правила та умови проектування містять ключові слова, що мають конкретні значення. До цих слів, що ґрунтуються на визначеннях у Запиті про надання коментарів – RFC 2119 “Key words for use in RFCs to indicate requirement level” (*«Ключові слова для використання в RFC з метою позначення рівнів вимог»*), опублікованому Інженерною радою Інтернет (Internet Engineering Task Force – IETF), належать слова:

- **ПОВИНЕН.** Це слово, або слова «НЕОБХІДНО» чи «СЛІД», означає, що визначення є безумовною вимогою специфікації;
- **НЕ ПОВИНЕН.** Цей вираз, або вираз «НЕ СЛІД», означає, що визначення є безумовним запереченням специфікації;
- **МАЄ.** Це слово, або дієприкметник «РЕКОМЕНДОВАНО», означає, що за певних обставин можуть існувати вагомі причини для ігнорування певного елементу, проте, перед вибором напрямку подальших дій необхідно ретельно зважити і проаналізувати всі наслідки дій, позначених цим словом;
- **НЕ МАЄ.** Цей вираз, або вираз «НЕ РЕКОМЕНДОВАНО», означає, що за певних обставин можуть існувати вагомі причини для того, аби вважати певний напрямок дій прийнятним чи навіть корисним, але необхідно ретельно зважити і проаналізувати всі наслідки дій, позначених цим виразом;
- **МОЖЕ.** Це слово, або прислівник «НЕОБОВ'ЯЗКОВО», означає, що елемент є справді необов'язковим. Розробка, яка не передбачає певної опції, **ПОВИННА** бути підготовлена до взаємодії з іншою розробкою, яка передбачає таку опцію, навіть, і скоріше за все, зі зниженою функціональністю. І навпаки, розробка, яка передбачає певну опцію **ПОВИННА** бути підготовлена до взаємодії з іншою розробкою, яка не передбачає цієї опції (крім, звісно, функції, яку надає така опція).

Однак, ці ж слова, написані маленькими буквами, вживаються у звичайних значеннях.

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;
- **Courier:** позначає ключові слова, імена і коди XML, відтворені шрифтом Courier.

1.5.3 Ідентифікатори правил

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

- (а) “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 послідовним номером правила.

Наприклад, ідентифікатор [GD-10] позначає десяте загальне правило проектування.

2. УМОВИ 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 «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. Ці угоди також можуть застосовуватися для присвоєння імен файлам.

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 термінів, а саме:

- 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 виражається визначником 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 для 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] Назви типів, елементів та атрибутів ПОВИННІ наводитися англійськими словами і відповідати вимогам до написання, зазначеним в Оксфордському словнику англійської мови (Oxford English Dictionary), включаючи відомчі назви, але за винятком скорочень, абrevіатур або інших слів, наведених у Доповненні С.
- [GD-05] Назви типів, елементів та атрибутів ПОВИННІ складатися лише з іменників, прикметників або дієслів у теперішньому часі.
- [GD-06] Символи для використання в назвах типів, елементів або атрибутів ПОВИННІ обмежуватися таким набором: {a-z, A-Z і 0-9}.
- [GD-07] Максимальна довжина назви МАЄ складати 35 символів.
- [GD-08] Назви типів, елементів та атрибутів МАЮТЬ бути стислими та очевидними.
- [GD-09] Назви елементів ПОВИННІ відповідати угоді Upper CamelCase (UCC). Наприклад, `CountryCode`.
- [GD-10] Назви типів ПОВИННІ відповідати угоді UCC + суфікс "Type". Наприклад, `ApplicantType`.
- [GD-11] Назви атрибутів ПОВИННІ відповідати угоді Lower CamelCase (LCC). Наприклад, `currencyCode="EUR"`.
- [GD-12] Скорочення та абrevіатури, наведені у Доповненні С, ПОВИННІ завжди використовуватися замість повних розширених назв.
- [GD-13] Скорочення та абrevіатури на початку оголошення атрибута ПОВИННІ наводитися у нижньому регістрі. Всі інші скорочення та абrevіатури в оголошенні атрибута ПОВИННІ наводитися у верхньому регістрі.
- [GD-14] Скорочення та абrevіатури для всіх назв елементів і типів ПОВИННІ наводитися у верхньому регістрі.
- [GD-15] Імена комплексних типів МАЮТЬ містити значущі об'єктні класи.
- [GD-16] Імена комплексних поєднаних типів МАЮТЬ використовувати структуру об'єктного класу комплексного типу, до якого приєднують, властивості (характеру поєднання) і об'єктного класу комплексного типу, що приєднується, а також будь-яких кваліфікаторів. Наприклад, у назві `ApplicantResidenceAddress`: `Applicant` є об'єктним класом комплексного типу, до якого приєднують, `Residence` є властивістю, а `Address` – об'єктний класом комплексного типу, що приєднується.
- [GD-17] Імена простих типів та атомарних елементів (без дочірніх складових) МАЮТЬ складатися з об'єктного класу, властивості, терміну представлення і кваліфікатора.
- [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] Об'єктний клас в імені ПОВИНЕН зазначатися на першому місці (ліворуч), властивість – на наступному, а термін представлення – на останньому (праворуч). Кваліфікатор МАЄ передувати об'єктному класу, що приєднується, або властивості.
- [GD-22] Якщо властивість закінчується тим самим (або ж еквівалентним) словом, що і термін представлення, то термін представлення ПОВИНЕН бути видалений.
- [GD-23] Терміни представлення, наведені у Доповненні В, ПОВИННІ використовуватися як терміни для представлення в іменах компонентів.
- [GD-24] Усі назви типів, елементів та атрибутів (оголошених локально чи глобально) у просторі імен ПОВИННІ бути унікальними.
- [GD-25] Слово або слова у назві МАЮТЬ зазначатися в однині, за винятком, якщо поняття передається лише множиною. Наприклад, `TotalMarkSeries`.
- [GD-26] Наприкінці назви сукупності МАЄ зазначатися суфікс “Bag”. Наприклад, `EmailAddressBag` означає сукупність `EmailAddress`.
- [GD-27] Сполучні слова на зразок сполучника “and”, прийменника “of” та артикля “the” НЕ МАЮТЬ використовуватися в назвах типів, елементів та атрибутів, за винятком, якщо вони складають частину виробничої термінології. Наприклад, `GoodsAndServices`.
- [GD-28] Назви типів, елементів та атрибутів НЕ ПОВИННІ перекладатися, змінюватися або замінюватися за жодних умов.
- [GD-29] Назви типів та елементів НЕ ПОВИННІ містити посилання на статті і номери 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. Угоди про іменування файлів схем є важливою частиною стратегії керування версіями схем. Назви файлів схем повинні присвоюватися послідовно, аби користувачі мали змогу розрізняти версії схем. Таким чином, правила, наведені у цьому розділі, тісно пов'язані з правилами, викладеними у розділі 2.6 «*Керування версіями схем*» цього документа.

16. Файл схеми має містити інформацію про її версію. Рекомендується до назви файла схеми включати ідентифікатор версії. До назв схем, що знаходяться на стадії проектування

(проекти схем), дозволяється додавати номери редакцій. Проекти схем повинні позначатися шляхом додавання літери "D" і номеру редакції до їх назв.

- [GD-30] Символи для використання в іменах файлів схем ПОВИННІ обмежуватися таким набором: {a-z, A-Z, 0-9, дефіс (-) і крапка (.)}.
- [GD-31] Ім'я файла схеми ПОВИННЕ складатися з п'яти частин і трьох роздільників, тобто: <назва компонента>"-"<номер основної версії>"-"<номер проміжної версії>"-"<розширення файла>. Наприклад, `EmailAddressType-V1-1.xsd`, `languageCode-V1-0.xsd`, `id-V1-2.xsd`.
- [GD-32] Ім'я файла проекту схеми ПОВИННЕ складатися з семи частин і чотирьох роздільників, тобто: <назва компонента>"-"<номер основної версії>"-"<номер проміжної версії>"-"<номер редакції>"-"<розширення файла>, наприклад, `Contact-V1-0-D3.xsd`.

2.3 Принцип модульної побудови

17. Стандарт BOIB ST.96 ґрунтується переважно на принципі модульної побудови під час проектування схем. Поділ схем на численні модулі забезпечує можливість повторного використання компонентів шляхом окремого їх імпортування, уникаючи необхідності імпортувати схеми повністю. Тому стандарт BOIB ST.96 рекомендує уникати визначення всіх елементів і логічних компонентів в одній суцільній схемі XML, що перешкоджає обміну та повторному використанню окремих елементів і логічних компонентів, визначених у схемі як група.

2.3.1 Модулі схем

18. Компоненти, визначені у стандарті BOIB ST.96, поділяються на загальні (*Common*) і спеціальні за типами ОПВ, тобто «Винахід / корисна модель» (*Patent*), «Торговельна марка» (*Trademark*) або «Промисловий зразок» (*Design*). Загальні компоненти МАЮТЬ бути нейтральними за контекстом (незалежними від виробничої діяльності) і загальнодоступними принаймні для компонентів двох типів ОПВ.

19. Існують три рівні компонентів: основний (*Basic*), сукупний (*Aggregate*) і документ (*Document*). Загальні компоненти поділяються на основний і сукупний рівні, а спеціальні - на основний, сукупний і документ.

Рівень	Опис
Компонент основного рівня	Компоненти основного рівня використовуються одним або декількома компонентами сукупного рівня. Компонент основного рівня представляється як елемент. Компонент основного рівня посилається на визначення вбудованого типу даних схеми W3C, простого типу або

	комплексного типу з визначенням <code>xsd:simpleContent</code>. Наприклад, <code>FirstName (string)</code>, <code>PhoneNumberCategory (PhoneNumberCategoryType)</code>.
Компонент сукупного рівня	Компонент сукупного рівня – це множина пов’язаних компонентів основного рівня та/або інших компонентів сукупного рівня, які разом мають очевидне значення для виробничої діяльності залежно або незалежно від певного контексту цієї діяльності. Він представляється як елемент і тип (комплексний тип, що включає низку елементів). Компонент сукупного рівня посилається на визначення компонента осново рівня, або іншого компонента сукупного рівня, або сторонні стандарти, зазначені в цьому документі. Наприклад, <code>Name (NameType)</code>, <code>Contact (ContactType)</code>.
Компонент рівня документ	Компонент рівня документ є компонентом найвищого рівня, що визначає основні документи для обміну. Він представляється як елемент і комплексний тип. Компонент рівня документ посилається на визначення компонента основного рівня, або компонента сукупного рівня, або сторонніх стандартів, зазначених у цьому документі. Наприклад, <code>ApplicationBody (ApplicationBodyType)</code>, <code>MarkRecord (MarkRecordType)</code>.

20. У цьому прикладі наведено оголошення для компонента основного рівня:

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

2.3.2 Зовнішні посилання у схемах

21. Кожний модуль схеми знаходиться у певному просторі імен. У цьому документі, кожний модуль схеми визначається в одному з чотирьох просторів імен, тобто, для модулів схем загального компонента, для модулів схем компонента «Винахід / корисна модель»,

модулів схем компонента «Торговельна марка» і для модулів схем компонента «Промисловий зразок».

22. 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.

23. На Рис. 1 показано використання модульної моделі за стандартом BOIV ST.96 для розробки схем.

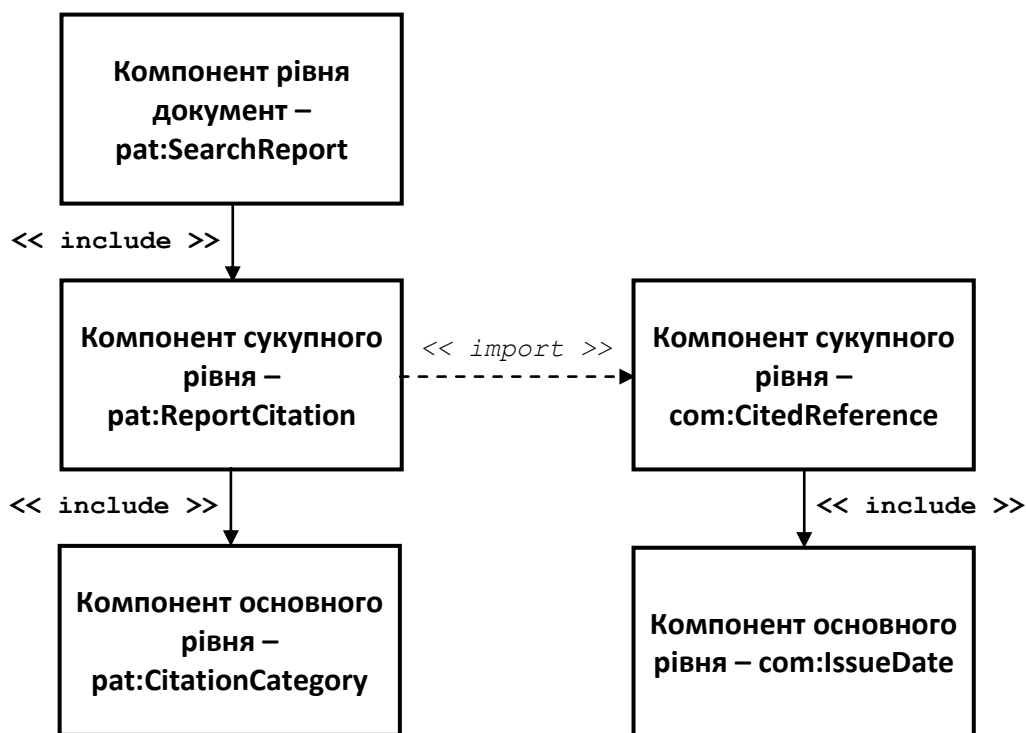


Рис. 1. Зразок модульної побудови схеми

[SD-01] Модулі схем для компонента «Винахід / корисна модель», модулі схем для компонента «Торговельна марка» і модулі схем для компонента «Промисловий зразок» ПОВИННІ використовувати структурний компонент `import` для посилання на модулі схем для загального компонента.

- [SD-02] Модулі схем загального компонента НЕ ПОВИННІ посилатися на модулі схем компонента «Винахід / корисна модель», модулі схем «Торговельна марка» і модулі схем компонента «Промисловий зразок».
- [SD-03] Модулі схем компонентів «Винахід / корисна модель», «Торговельна марка» і «Промисловий зразок» не повинні посилатися один на одного.

2.4 *Можливість повторного використання*

24. 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.

25. 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.

26. 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.

27. 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 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 “Garden of Eden”.

- [SD-04] Якщо можливо, ПОВИННІ використовуватися існуючі схеми, а не створюватися нові.
- [SD-05] Схеми МАЮТЬ максимально використовувати існуючі елементи і типи.
- [SD-06] Усі типи, елементи та атрибути ПОВИННІ оголошуватися глобально.

- [SD-07] Схеми НЕ ПОВИННІ використовувати структурний компонент `redefine` для повторного визначення.

2.5 Простори імен

28. 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 URL.
- [SD-13] Усі схеми ПОВИННІ мати атрибути `elementFormDefault` і `attributeFormDefault` зі значенням `"qualified"`.

29. 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

30. Присвоєння назв просторам імен є важливою складовою стратегії керування версіями схем. Під час проектування схем необхідно беззаперечно *дотримуватися* правил присвоєння назв просторам імен. Незважаючи на наявність номера проміжної версії певної схеми, назва простору імен повинна містити лише номер основної версії. Наприклад, <http://www.wipo.int/standards/XMLSchema/Trademark/1>.

31. Належно сформований простір імен наведено в наступному прикладі:

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

32. Докладна інформація міститься в розділі «Керування версіями схем» цього документа.

[SD-14] Назви просторів імен ПОВИННІ мати таку структуру: <http://www.wipo.int/standards/XMLSchema/<category>/<major>>, де <category> – це ідентифікатор домену схеми, що використовується: Common, Patent, Trademark або Design; а <major> – це номер основної версії.

[SD-15] Схеми МАЮТЬ використовувати "xsd" як префікс простору імен для всіх структурних компонентів схеми W3C, "com" для всіх схем загального компонента, "pat" для всіх схем компонента «Винахід / корисна модель», "tmk" для всіх схем компонента «Торговельна марка», "dgn" для всіх схем компонента «Промисловий зразок», "tbl" для схем табличного компонента OASIS і "mathml" для схем MathML.

2.5.3 Цільові простори імен

33. 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.

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

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

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

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

- [SD-16] Кожна схема ПОВИННА містити оголошення заданого простору імен з використанням атрибута `xsd:targetNamespace`.
- [SD-17] Атрибут схеми `targetNamespace` ПОВИНЕН відповідати назві простору імен одного з класифікаторів оголошеного простору імен, окрім класифікатора простору імен схеми W3C.
- [SD-18] Схеми для загальних компонентів ПОВИННІ використовувати <http://www.wipo.int/standards/XMLSchema/Common/1> як цільовий простір імен.
- [SD-19] Схеми для компонента «Винахід / корисна модель» ПОВИННІ використовувати <http://www.wipo.int/standards/XMLSchema/Patent/1> як цільовий простір імен.
- [SD-20] Схеми для компонента «Торговельна марка» ПОВИННІ використовувати <http://www.wipo.int/standards/XMLSchema/Trademark/1> як цільовий простір імен.
- [SD-21] Схеми для компонента «Промисловий зразок» ПОВИННІ використовувати <http://www.wipo.int/standards/XMLSchema/Design/1> як цільовий простір імен.

2.5.4 Простори імен за замовчуванням

35. 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] Схема HE MAЄ використовувати простір імен за замовчуванням.

2.6 Керування версіями схем

36. У випадку, 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

37. 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

38. 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;
- видаляння або додавання обов'язкових елементів чи атрибутів;
- змінювання множини, тобто необов'язкової на обов'язкову.

2.6.1.2 Проміжні версії

39. 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 у випадку:

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

- [SD-23] Нові проміжної версії схем ПОВИННІ забезпечувати відповідність усіх екземплярів, створених на підставі попередніх проміжних версій схем, одній основній версії.
- [SD-24] Якщо новий проект схеми не забезпечує відповідність існуючого екземпляра XML, створеного на підставі поточної основної версії, номер основної версії схеми ПОВИНЕН бути збільшений.
- [SD-25] Номери проміжних версій схем НЕ МАЮТЬ збільшуватися, окрім випадків, коли ці схеми або одна з імпортованих/включених схем не зазнають незначних змін.
- [SD-26] Ім'я файла схеми, атрибут `version` схеми W3C і простір імен ПОВИННІ містити однакову інформацію стосовно версії схеми.
- [SD-27] У випадку оновлення включеної або імпортованої схеми, номер версії схеми МАЄ бути змінений.
- [SD-28] Якщо будь-який структурний компонент схеми зазнає значних змін у заданому просторі імен, номери версій усіх схем у цьому просторі імен 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

40. Визначення стратегії керування версіями схем передбачає одночасну взаємодію наступних складових:

- а) Угода про іменування просторів імен схем;
- б) Угода про іменування файлів схем;
- в) використання вбудованого атрибут `version` схеми XML;
- г) використання атрибут `schemaVersion`, що визначається користувачем.

2.6.2.1 Простори імен у межах керування версіями схем

41. Відповідно до правил іменування просторів імен, визначених у цьому документі, простори імен мають містити лише номер основної версії. Це є важливим для забезпечення сумісності між проміжними версіями схеми. Оскільки файли попередніх проміжних версій повинні відповідати новим проміжним версіям схеми, простір імен повинен зберігатися у всіх проміжних версіях схем.

42. Сам по собі простір імен не може використовуватися для однозначної ідентифікації точної версії відповідної схеми або екземпляра документа. Для цього використовуються інші важливі особливості керування версіями схем, описані нижче.

2.6.2.2 Угоди про іменування файлів у межах керування версіями схем

43. Інформація про версію схеми може бути включена до назви її файла. Відповідно до правил іменування, визначених у цьому документі, ім'я файла схеми має містити як номер основної, так і номер проміжної версії.

2.6.2.3 Вбудований у схему XML атрибут *version* у межах керування версіями схем

44. 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/Patent/1"
targetNamespace="http://www.wipo.int/standards/XMLSchema/Patent/1"
elementFormDefault="qualified" attributeFormDefault="unqualified" version="1.0">
...
</xsd:schema>
```

[SD-30] Атрибут *version* схеми W3C ПОВИНЕН включати як номер основної, так і номер проміжної версії для кожного файла схеми у наступному форматі:
<номер основної версії>.<номер проміжної версії>.

2.6.2.4 Атрибут *schemaVersion*, що визначається користувачем, у межах керування версіями схем для екземплярів XML

45. Атрибут *schemaVersion*, що визначається користувачем, необхідний для того, щоб переконатися, що даний екземпляр документа відноситься безпосередньо до конкретної версії схеми. Цей атрибут чітко визначає версію схеми, на яку посилається екземпляр документа. Наприклад:

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

Відповідний екземпляр XML:

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

- [SD-31] Схеми, які визначаються як компоненти рівня документ, **ПОВИННІ** містити визначення обов'язкового атрибута з іменем `schemaVersion` у своєму корінному елементі в наступному форматі: <номер основної версії>.<номер проміжної версії>.

2.7 Взаємоперетворення з іншими стандартами BOIB, що стосуються XML

46. До прийняття стандарту BOIB ST.96, ВПВ користувалися стандартами BOIB ST.36, ST.66 і ST.86. Тому, підтримка сумісності з існуючими документами, створеними за стандартами BOIB ST.36, ST.66 і ST.86, є одним з 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.

47. 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*.

48. Переліки перерахування 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 повинні 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).

- [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 у схемах **НЕ МАЮТЬ** включати значення «Інше» або «Невизначено», якщо відсутня необхідність.

2.8 Схеми промислових стандартів

¹ Тут «кардинальність» означає кількість елементів у наборі.

49. Якщо це вимагається змістом документа, тобто, якщо зміст не є унікальним для сфери промислової власності, потрібно використовувати схеми промислових стандартів. Цей документ рекомендує використовувати схеми таких промислових стандартів: таблична схема XML OASIS і схема MathML. 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. Для цілей сфери промислової власності, схема OASISTable.xsd визначається у межах схеми за стандартом BOIB ST.96 як типова. 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>.

50. 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 використовуватися схема MathML версії 3.
- [SD-35] Для таблиць МАЄ використовуватися схема OASISTable.xsd, створена на основі схеми XML OASIS.
- [SD-36] Схеми промислових стандартів МАЮТЬ включатися лише за допомогою посилання і після попереднього погодження Комітету зі стандартів BOIB.

3. УМОВИ ПОБУДОВИ СХЕМ XML

3.1 *Визначення типів*

51. 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

52. 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

53. 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

54. Однією з переваг, яку надають схеми XML, є можливість визначати власні типи 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] Прості типи МАЮТЬ використовуватися для визначення списків перелічуваних кодів.
- [SD-39] Прості типи МОЖУТЬ використовувати `xsd:union` для розширення списку кодів.
- [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] Значення дати і часу в екземплярах схеми XML ПОВИННІ відповідати типам даних часу і дати схеми W3C.
- [SD-44] Для кодування валют ПОВИННІ використовуватися коди за стандартом ISO 4217-Alpha (трибуквені коди валют).
- [SD-45] Символи для використання в переліках ПОВИННІ обмежуватися таким набором: {a-z, A-Z, 0-9, крапка (.), кома (,), пробіл, дефіс (-), амперсанд (&);) і підкреслювання (_)}.
- [SD-46] Значення, доступні у переліках, МАЮТЬ бути «семантично достатніми» і наводитися англійською мовою з якомога меншою кількістю символів. Значення МАЮТЬ походити із загальноприйнятої термінології у сфері промислової власності.

3.1.2 Комплексні типи

55. Комплексні типи – це типи, що визначаються користувачем і містять підрядні елементи та/або атрибути.

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

3.2 Елементи та атрибути

56. 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

57. 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. 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] У схемах МАЮТЬ використовуватися атрибути лише для визначення невиробничих даних. Наприклад, атрибут `sequenceNumber`.

3.2.2 Елементи

3.2.2.1 Кардинальність елементів

58. Термін «кардинальність» визначається як кількість елементів у наборі.

59. 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] У схемах НЕ МАЮТЬ використовуватися показники появи для значень за замовчуванням (тобто, `minOccurs="1"; maxOccurs="1"`).

3.2.2.2 Порожні елементи

60. Порожній елемент – це елемент без 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.

61. 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

62. 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 поняття «підрядний атрибут» не існує. Отже, атрибути не можуть бути розширені на зразок елементів.

63. Кардинальність атрибутів відрізняється від кардинальності елементів. Для атрибута може бути визначений показник `use` з одним із таких значень: `"required"`, `"optional"` або `"prohibited"`.

[SD-51] Показник `use` НЕ МАЄ використовуватися для атрибутів у схемах, якщо цільове значення є за замовчуванням (тобто, `use="optional"`).

3.2.4 Групування елементів та атрибутів

64. Формувальники – це структурні компоненти схеми W3C, що угруповують оголошення елементів. Стандарт схеми W3C передбачає три типи формувальників, а саме, `sequence`, `choice` та `all`.

65. 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.

66. Формувальник `choice` вказує на те, що лише один елемент, оголошений в ньому, може наводитися в екземплярі XML.

67. 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.

68. 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 необов'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

69. 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

70. Розширення комплексних типів не впливає на гармонізацію даних на відміну від розширення простих типів, оскільки компоненти, що додаються, не підлягають обміну; 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

71. З точки зору гармонізації даних, обмеження комплексних типів впливає на структуру і може призвести до проблем із сумісністю, що перешкоджатиме процесу обміну даними. У випадку простих типів, обмеження впливає на значення, що не передбачає появи проблем під час обміну даними.

3.3.3 Групи заміщення

72. Групи заміщення надають можливість замінити один глобальний елемент в екземплярі XML іншим глобальним елементом без необхідності додаткового коригування схеми. Однак, групи заміщення не забезпечуються гармонізацією 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

73. 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:

- `xsd:ID/xsd:IDREF/IDREFS`, які безпосередньо імітують атрибути типів `ID`, `IDREF` та `IDREFS` зі схем XML DTD;
- `xsd:key/xsd:keyref/xsd:unique`, які були запроваджені для забезпечення більшої гнучкості завдяки використанню виразів XPath.

[SD-56] Для обмеження ідентифікації у схемах МАЮТЬ використовуватися `xsd:key/xsd:keyref/xsd:unique`.

[SD-57] `ID/IDREF/IDREFS` НЕ МАЮТЬ використовуватися у схемах для обмеження ідентифікації.

3.5 Документування схем

74. Специфікація схеми W3C дозволяє включати до схем документування. Це дозволяє схемам містити вбудовані описи для кожного структурного компонента. Документування необхідне для роз'яснення змісту та способу використання елементів у схемах користувачам, які не володіють такими ж знаннями або не мають такої ж кваліфікації, що і розробник схем.

75. 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.

76. 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 документування схеми W3C. 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.

3.5.1 Документування заголовків схем

77. 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. Як і для документації структурних компонентів схеми, для документації в заголовку схеми застосовується елемент документації схеми W3C.

78. Таблиця 1 містить елементи, які повинні включатися до заголовків усіх схем.

Таблиця 1. Інформація у заголовку

Назва елементу в заголовку	Опис
"Last update"	Дата останнього оновлення схеми
"Contact Point"	Адреса контактної особи для вирішення питань стосовно схеми
"Release Notes URL"	URL-адреса, за якою опубліковано інформацію стосовно версії схеми.

Назва елементу заголовку має передувати його змісту, як показано у наступному прикладі:

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

79. 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 ПОВИННІ містити документування у заголовках з елементами “Last update”, “Contact point” і “Release notes URL”.

4. ПРАВИЛА ПРОЕКТУВАННЯ ЕКЗЕМПЛЯРІВ XML

80. 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

81. 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.

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

83. 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/Common/1"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://www.wipo.int/standards/XMLSchema/Common/1/Aggregate/Contact-V1-0.xsd">
.....
</com:Contact>
```

84. 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/Common/1>" є заданим простором імен схеми "Contact.xsd".

- [ID-01] Екземпляри XML ПОВИННІ бути перевірені на відповідність певній схемі.
- [ID-02] В екземплярах документа XML ПОВИНЕН використовуватися атрибут `schemaLocation` для опису простору імен і назви 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).

85. 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.

86. 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.

87. 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. Таким чином, 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 перемикачання у схемі, що використовує два показники:

- `elementFormDefault`,
- `attributeFormDefault`.

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

89. 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 «Винахід / корисна модель», префікс "tmk" для всіх структурних компонентів схеми компонента «Торговельна марка», префікс "dgn" для всіх структурних компонентів схеми компонента «Промисловий зразок», 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

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

91. 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

92. Простори імен в екземплярах документів XML мають видимість охоплення щодо застосування у екземплярах XML-документів. Видимість простору імен застосовується для оголошеного елементу (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.

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

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

4.2 Зовнішні сутності

94. 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.

95. Екземпляр XML може містити посилання на інший файл 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.

96. 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 вірус.

- [ID-08] Зовнішні сутності, які є зображеннями, 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 A-D наведено далі]

ДОПОВНЕННЯ А – ЗВЕДЕНИЙ 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. Unicode». Символи Unicode ПОВИННІ кодуватися в UTF-8.
[GD-04]	Назви типів, елементів та атрибутів ПОВИННІ наводитися англійськими словами і відповідати вимогам до написання, зазначеним в Оксфордському словнику англійської мови (Oxford English Dictionary), включаючи відомчі назви, але за винятком скорочень, аббревіатур або інших слів, наведених у Доповненні С.
[GD-05]	Назви типів, елементів та атрибутів ПОВИННІ складатися лише з іменників, прикметників або дієслів у теперішньому часі.
[GD-06]	Символи для використання в назвах типів, елементів або атрибутів ПОВИННІ обмежуватися таким набором: {a-z, A-Z і 0-9}.
[GD-07]	Максимальна довжина назви МАЄ складати 35 символів.
[GD-08]	Назви типів, елементів та атрибутів МАЮТЬ бути стислими та очевидними.
[GD-09]	Назви елементів ПОВИННІ відповідати угоді Upper CamelCase (UCC). Наприклад, CountryCode.
[GD-10]	Назви типів ПОВИННІ відповідати угоді UCC + суфікс "Type". Наприклад, ApplicantType.
[GD-11]	Назви атрибутів ПОВИННІ відповідати угоді Lower CamelCase (LCC). Наприклад, currencyCode="EUR".
[GD-12]	Скорочення та аббревіатури, наведені у Доповненні С, ПОВИННІ завжди використовуватися замість повних розширених назв.
[GD-13]	Скорочення та аббревіатури на початку оголошення атрибута ПОВИННІ наводитися у нижньому регістрі. Всі інші скорочення та аббревіатури в оголошенні атрибута ПОВИННІ наводитися у верхньому регістрі.
[GD-14]	Скорочення та аббревіатури для всіх назв елементів і типів ПОВИННІ

	наводитися у верхньому реєстрі.
[GD-15]	Імена комплексних типів МАЮТЬ містити значущі об'єктні класи.
[GD-16]	Імена комплексних поєднаних типів МАЮТЬ використовувати структуру об'єктного класу комплексного типу, до якого приєднують, властивості (характеру поєднання) і об'єктного класу комплексного типу, що приєднується, а також будь-яких кваліфікаторів. Наприклад, у назві <code>ApplicantResidenceAddress: Applicant</code> є об'єктним класом комплексного типу, до якого приєднують, <code>Residence</code> є властивістю, а <code>Address</code> – об'єктний класом комплексного типу, що приєднується.
[GD-17]	Імена простих типів та атомарних елементів (без дочірніх складових) МАЮТЬ складатися з об'єктного класу, властивості, терміну представлення і кваліфікатора.
[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]	Об'єктний клас в імені ПОВИНЕН зазначатися на першому місці (ліворуч), властивість – на наступному, а термін представлення – на останньому (праворуч). Кваліфікатор МАЄ передувати об'єктному класу, що приєднується, або властивості.
[GD-22]	Якщо властивість закінчується тим самим (або ж еквівалентним) словом, що і термін представлення, то термін представлення ПОВИНЕН бути видалений.
[GD-23]	Терміни представлення, наведені у Доповненні В, ПОВИННІ використовуватися як терміни для представлення в іменах компонентів.
[GD-24]	Усі назви типів, елементів та атрибутів (оголошених локально чи глобально) у просторі імен ПОВИННІ бути унікальними.
[GD-25]	Слово або слова у назві МАЮТЬ зазначатися в однині, за винятком, якщо поняття передається лише множиною. Наприклад, <code>TotalMarkSeries</code> .
[GD-26]	Наприкінці назви сукупності МАЄ зазначатися суфікс “Bag”. Наприклад, <code>EmailAddressBag</code> означає сукупність <code>EmailAddress</code> .
[GD-27]	Сполучні слова на зразок сполучника “and”, прийменника “of” та артикля “the” НЕ МАЮТЬ використовуватися в назвах типів, елементів та атрибутів,

	за винятком, якщо вони складають частину виробничої термінології. Наприклад, <code>GoodsAndServices</code> .
[GD-28]	Назви типів, елементів та атрибутів НЕ ПОВИННІ перекладатися, змінюватися або замінюватися за жодних умов.
[GD-29]	Назви типів та елементів НЕ ПОВИННІ містити посилання на статті і номери правил. Наприклад, <code>PCTRule702C</code> .
[GD-30]	Символи для використання в іменах файлів схем ПОВИННІ обмежуватися таким набором: {a-z, A-Z, 0-9, дефіс (-) і крапка (.)}.
[GD-31]	Ім'я файла схеми ПОВИННЕ складатися з п'яти частин і трьох роздільників, тобто: <назва компонента>"-"<"V"><номер основної версії>"-"<номер проміжної версії>"."<розширення файла>. Наприклад, <code>EmailAddressType-V1-1.xsd</code> , <code>languageCode-V1-0.xsd</code> , <code>id-V1-2.xsd</code> .
[GD-32]	Ім'я файла проекту схеми ПОВИННЕ складатися з семи частин і чотирьох роздільників, тобто: <назва компонента>"-"<"V"><номер основної версії>"-"<номер проміжної версії>"-"<"D"><номер редакції>"."<розширення файла>, наприклад, <code>Contact-V1-0-D3.xsd</code> .

Правила проектування схем

Ідентифікатор	Правило
[SD-01]	Модулі схем для компонента «Винахід / корисна модель», модулі схем для компонента «Торговельна марка» і модулі схем для компонента «Промисловий зразок» ПОВИННІ використовувати структурний компонент <code>import</code> для посилання на модулі схем для загального компонента.
[SD-02]	Модулі схем загального компонента НЕ ПОВИННІ посилатися на модулі схем компонента «Винахід / корисна модель», модулі схем «Торговельна марка» і модулі схем компонента «Промисловий зразок».
[SD-03]	Модулі схем компонентів «Винахід / корисна модель», «Торговельна марка» і «Промисловий зразок» не повинні посилатися один на одного.
[SD-04]	Якщо можливо, ПОВИННІ використовуватися існуючі схеми, а не створюватися нові.
[SD-05]	Схеми МАЮТЬ максимально використовувати існуючі елементи і типи.
[SD-06]	Усі типи, елементи та атрибути ПОВИННІ оголошуватися глобально.
[SD-07]	Схеми НЕ ПОВИННІ використовувати структурний компонент <code>redefine</code> для

	повторного визначення.
[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 URL.
[SD-13]	Усі схеми ПОВИННІ мати атрибути <code>elementFormDefault</code> і <code>attributeFormDefault</code> зі значенням "qualified".
[SD-14]	Назви просторів імен ПОВИННІ мати таку структуру: <a href="http://www.wipo.int/standards/XMLSchema/<category>/<major>">http://www.wipo.int/standards/XMLSchema/<category>/<major> , де <code><category></code> – це ідентифікатор домену схеми, що використовується: Common, Patent, Trademark або Design; а <code><major></code> – це номер основної версії.
[SD-15]	Схеми МАЮТЬ використовувати "xsd" як префікс простору імен для всіх структурних компонентів схеми W3C, "com" для всіх схем загального компонента, "pat" для всіх схем компонента «Винахід / корисна модель», "tmk" для всіх схем компонента «Торговельна марка», "dgn" для всіх схем компонента «Промисловий зразок», "tbl" для схем табличного компонента OASIS і "mathml" для схем MathML.
[SD-16]	Кожна схема ПОВИННА містити оголошення заданого простору імен з використанням атрибута <code>xsd:targetNamespace</code> .
[SD-17]	Атрибут схеми <code>targetNamespace</code> ПОВИНЕН відповідати назві простору імен одного з класифікаторів оголошеного простору імен, окрім класифікатора простору імен схеми W3C.
[SD-18]	Схеми для загальних компонентів ПОВИННІ використовувати <a href="http://www.wipo.int/standards/XMLSchema/Common/<major>">http://www.wipo.int/standards/XMLSchema/Common/<major> як цільовий простір імен.
[SD-19]	Схеми для компонента «Винахід / корисна модель» ПОВИННІ використовувати <a href="http://www.wipo.int/standards/XMLSchema/Patent/<major>">http://www.wipo.int/standards/XMLSchema/Patent/<major> як цільовий простір імен.
[SD-20]	Схеми для компонента «Торговельна марка» ПОВИННІ використовувати <a href="http://www.wipo.int/standards/XMLSchema/Trademark/<major>">http://www.wipo.int/standards/XMLSchema/Trademark/<major> як цільовий простір імен.
[SD-21]	Схеми для компонента «Промисловий зразок» ПОВИННІ використовувати <a href="http://www.wipo.int/standards/XMLSchema/Design/<major>">http://www.wipo.int/standards/XMLSchema/Design/<major> як цільовий

	простір імен.
[SD-22]	Схема HE МАЄ використовувати простір імен за замовчуванням.
[SD-23]	Нові проміжної версії схем ПОВИННІ забезпечувати відповідність усіх екземплярів, створених на підставі попередніх проміжних версій схем, одній основній версії.
[SD-24]	Якщо новий проект схеми не забезпечує відповідність існуючого екземпляра XML, створеного на підставі поточної основної версії, номер основної версії схеми ПОВИНЕН бути збільшений.
[SD-25]	Номери проміжних версій схем НЕ МАЮТЬ збільшуватися, окрім випадків, коли ці схеми або одна з імпортованих/включених схем не зазнають незначних змін.
[SD-26]	Ім'я файлу схеми, атрибут <code>version</code> схеми W3C і простір імен ПОВИННІ містити однакову інформацію стосовно версії схеми.
[SD-27]	У випадку оновлення включеної або імпортованої схеми, номер версії схеми МАЄ бути змінений.
[SD-28]	Якщо будь-який структурний компонент схеми зазнає значних змін у заданому просторі імен, номери версій усіх схем у цьому просторі імен 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 включати як номер основної, так і номер проміжної версії для кожного файлу схеми у наступному форматі: <номер основної версії>.<номер проміжної версії>.
[SD-31]	Схеми, які визначаються як компоненти рівня документ, ПОВИННІ містити визначення обов'язкового атрибута з іменем <code>schemaVersion</code> у своєму корінному елементі в наступному форматі: <номер основної версії>.<номер проміжної версії>.
[SD-32]	Схеми за стандартом BOIB ST.96 МАЮТЬ розроблятися 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 у схемах НЕ МАЮТЬ включати значення «Інше» або «Невизначено», якщо відсутня необхідність.
[SD-34]	Для математичних формул МАЄ використовуватися схема MathML версії 3.

⁵ Тут «кардинальність» означає кількість елементів у наборі.

[SD-35]	Для таблиць MAE використовуватися схема OASISTable.xsd, створена на основі схеми XML OASIS.
[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]	Прості типи МАЮТЬ використовуватися для визначення списків перелічуваних кодів.
[SD-39]	Прості типи МОЖУТЬ використовувати <code>xsd:union</code> для розширення списку кодів.
[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]	Значення дати і часу в екземплярах схеми XML ПОВИННІ відповідати типам даних часу і дати схеми W3C.
[SD-44]	Для кодування валют ПОВИННІ використовуватися коди за стандартом ISO 4217-Alpha (трибуквені коди валют).
[SD-45]	Символи для використання в переліках ПОВИННІ обмежуватися таким набором: {a-z, A-Z, 0-9, крапка (.), кома (,), пробіл, дефіс (-), амперсанд (&); і підкреслювання (_)}.
[SD-46]	Значення, доступні у переліках, МАЮТЬ бути «семантично достатніми» і наводитися англійською мовою з якомога меншою кількістю символів. Значення МАЮТЬ походити із загальноприйнятої термінології у сфері промислової власності.
[SD-47]	МОЖУТЬ використовуватися абстрактні комплексні типи.
[SD-48]	У схемах МАЮТЬ використовуватися атрибути лише для визначення невиробничих даних. Наприклад, атрибут <code>sequenceNumber</code> .
[SD-49]	У схемах НЕ МАЮТЬ використовуватися покажчики появи для значень за замовчуванням (тобто, <code>minOccurs="1";maxOccurs="1"</code>).

[SD-50]	Порожні елементи НЕ ПОВИННІ визначатися у схемах, окрім як для розривання рядків.
[SD-51]	Показник <code>use</code> НЕ МАЄ використовуватися для атрибутів у схемах, якщо цільове значення є за замовчуванням (тобто, <code>use="optional"</code>).
[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]	Для обмеження ідентифікації у схемах МАЮТЬ використовуватися <code>xsd:key/xsd:keyref/xsd:unique</code> .
[SD-57]	<code>ID/IDREF/IDREFS</code> НЕ МАЮТЬ використовуватися у схемах для обмеження ідентифікації.
[SD-58]	Усі схеми МАЮТЬ включати документування структурних компонентів схеми, використовуючи елемент документування схеми W3C. 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 ПОВИННІ містити документування у заголовках з елементами "Last update", "Contact point" і "Release notes URL".

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 опису простору імен і назви 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 "com" для всіх структурних компонентів схеми загального компонента, префікс "pat" для всіх структурних компонентів схеми компонента «Винахід / корисна модель», префікс "tmk" для всіх структурних компонентів схеми компонента «Торговельна марка», префікс "dgn" для всіх структурних компонентів схеми компонента «Промисловий зразок», 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.
[ID-06]	Екземпляри документів XML МАЮТЬ використовувати префікс xsi 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 сутності, які є зображеннями, 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.

[Кінець Доповнення А]
[Доповнення В наведено далі]

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 має певне значення. За допомогою цієї комбінації виражаються обмежені, попередньо визначені значення.	xsd:token
Дата	Позначення певного моменту у часі, представлене роком, місяцем та днем.	DateType
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	Цифра або рядок цифр, за допомогою яких виражається мітка, значення, кількість або ідентифікація.	xsd:positiveInteger або xsd:string
Відсоток	Число, за допомогою якого виражається частина цілого, поділена на 100.	xsd:decimal
Кількість	Кількість є зліченим числом негрошових одиниць, включаючи, можливо, дробі. QuantityType	QuantityType

	використовується для позначення зліченого числа предметів. <code>QuantityType</code> має використовуватися для простих властивостей об'єкта, що тлумачиться як комбінація, зібрання або контейнер для визначення кількості або підрахування його складових. <code>QuantityType</code> має завжди представляти собою злічене число предметів, а властивість може позначатися як «всього», «відправлено», «завантажено», «зберігається» тощо.	
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>
Час	Позначення певного хронологічного моменту в певний період.	<code>xsd:time</code>
Часова мітка	Зареєстрована дата і час, коли сталася подія.	<code>xsd:datetime</code>
URI	Уніфікований ідентифікатор ресурсів, що ідентифікує місцезнаходження файлу.	<code>xsd:anyURI</code>

[Кінець Доповнення В]

[Доповнення С наведено далі]

ДОПОВНЕННЯ С – ПЕРЕЛІК АБРЕВІАТУР І СКОРОЧЕНЬ

BOIB	Всесвітня організація інтелектуальної власності
ВПВ	Відомство промислової власності
МПК	Міжнародна патентна класифікація
НПЛ	Непатентна література
ОПВ	Об'єкт промислової власності
ПВ	Промислова власність
ALT	Альтернативний текст для зображення
B	Грубий шрифт
BioDeposit	Біологічне депонування
Br	Розрив рядка
DD	Опис визначення
DL	Список визначень
DOI	Цифровий ідентифікатор об'єкта
DT	Визначення
DTD	Визначення типу документа
I	Курсивний шрифт
ID	Ідентифікатор для системної ідентифікації
IDREF	Посилання на ідентифікатор
ISO	Міжнародна організація зі стандартизації
LCC	Угода Lower CamelCase
LI	Елемент списку
O	Закреслений текст
OCR	Оптичне розпізнавання символів
OL	Упорядкований список
P	Параграф
Pre	Попередньо відформатований текст

Sub	Підрядковий текст
Sup	Надрядковий текст
SWIFT	Товариство всесвітніх міжбанківських фінансових телекомунікацій
U	Підкреслений шрифт
UCC	Угода Upper CamelCase
UL	Невпорядкований список
URI	Уніфікований ідентифікатор ресурсу
URL	Уніфікований локатор ресурсу
URN	Уніфікована назва ресурсу
W3C	Консорціум Всесвітньої павутини
XML	Розширювана мова розмітки

[Кінець Доповнення С]

[Доповнення D наведено далі]

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.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, редакція 3: <http://www.w3.org/TR/MathML3>.

[Кінець Доповнення D і Додатка I]