

# Мова Службового Моделювання, Версія 1.1

## Рекомендації W3C від 12 травня 2009 року

### Ця версія:

<http://www.w3.org/TR/2009/REC-sml-20090512/>

### Остання версія:

<http://www.w3.org/TR/sml/>

### Попередня версія:

<http://www.w3.org/TR/2009/PR-sml-20090212/>

### Редактори:

Балхандра Пандіт, Корпорація Microsoft

Валентина Попески, Корпорація IBM

Вірджинія Кузнец, HP

Будь ласка, зверніться до [errata](#) цього документа, документ може включати в себе деякі нормативні виправлення.

Цей документ також доступний в таких ненормативних форматах як [XML](#).

Також дивись [translations](#).

Авторські права © 2009 W3C® ([MIT](#), [ERCIM](#), [Keio](#)), Всі права захищені. Застосовуються W3C правила відповідальності, документ використовує товарні знаки.

## Анотація

Дана специфікація визначає Мову Службового Моделювання, версія 1.1 (MCM), що використовується для моделювання складних послуг і систем, включаючи їх структуру, обмеження, політику і передову практику. MCM використовує XML схеми і Schematron.

## Статус цього Документа

Цей розділ описує статус цього документа на момент його публікації. Інші документи можуть замінити цей документ. Зі списком поточних W3C публікацій і останньою ревізією цієї технічної доповіді можна ознайомитися в [W3C технічного показнику доповідей](#) на <http://www.w3.org/TR/>.

12 травня 2009 Мова Службового Моделювання, версія 1.1 рекомендована W3C. Цей документ був розроблений [робочою групою Мови Службового Моделювання \(MCM\)](#), яка є частиною діяльності [Розширеної Мови Розмітки \(RPM\)](#).

[Громадський архів](#) Робочої групи містить схвальні відгуки та коментарі щодо даного документа. Отримані відгуки доступні.

Дизайн MCM було ретельно сплановано, він задовольняє усі технічні вимоги Робочої групи. З запропонованих 12 лютого 2009 рекомендацій були зроблені лише деякі редакційні зміни.

Цей документ був розглянутий членами W3C, розробниками програмного забезпечення,

а також іншими групами W3C і зацікавленими сторонами, був схвалений Директором в якості W3C рекомендації. Даний стабільний документ може бути використан в якості довідкового матеріалу або бути цитован в інших документах. Роль W3C у розробці даної рекомендації полягає в залученні уваги до цієї специфікації і в сприянні її широкого розповсюдження. Це підвищує функціональність і можливість взаємодії в Інтернеті.

Цей документ був підготовлений групою, що діє в рамках з [5 лютого 2004 W3C патентної політики](#). W3C підтримує [публічний список будь-яких патентів](#), зроблених у зв'язку з результатами роботи групи; ця сторінка містить інструкції для розкриття патентів. Будь-яка особа, яка зіткнулась з патентом, впевнена, що він містить основні претензії і повинен розкривати інформацію відповідно до [розділу 6 W3C з патентної політики](#).

## Зміст

1. Вступ (ненормативний)
2. Позначення і термінологія
  - 2.1 Умовні позначення
  - 2.2 Термінологія
  - 2.3 Простори імен MCM
3. Залежність від інших специфікацій
4. Посилання MCM
  - 4.1 Визначення посилання MCM
    - 4.1.1 Посилання MCM
    - 4.1.2 Недійсні посилання MCM
    - 4.1.3 Невирішені посилання MCM
    - 4.1.4 Мета посилання MCM
  - 4.2 Семантика посилання MCM
    - 4.2.1 Не більше однієї мети
    - 4.2.2 Послідовні посилання
    - 4.2.3 Ідентична мета
    - 4.2.4 Множинні посилання
    - 4.2.5 Недійсні посилання MCM
    - 4.2.6 Детермінована оцінка обмежень MCM
    - 4.2.7 smlfn: deref () - Функція розширення XPath
  - 4.3 Схеми посилань MCM
    - 4.3.1 URI посилання схеми MCM
      - 4.3.1.1 smlxpath1 () схема
5. Обмеження MCM
  - 5.1 Обмеження на посилання MCM
    - 5.1.1 SML: ациклічні
      - 5.1.1.1 Обмежені конструкції MCM
      - 5.1.1.2 Схема компонентів правил
      - 5.1.1.3 інстанція дійсності правил
    - 5.1.2 Обмеження на меті посилання MCM
      - 5.1.2.1 Обмеження конструкцій MCM
      - 5.1.2.2 Схема компонентів правил
      - 5.1.2.3 інстанції дії правил
    - 5.1.3 Короткий виклад обмежень посилань MCM (ненормативний)
  - 5.2 Ідентичність обмежень MCM

- 5.2.1 Синтаксис і семантика
  - 5.2.1.1 Конструкції обмежень MCM
  - 5.2.1.2 Правила компонентів схеми
  - 5.2.1.3 інстанції дії правил
- 5.3 Чинні обмеження значень обмежень MCM
- 5.4 Обмеження MCM і висновки комплексного типу
  - 5.4.1 Огляд обробки обмеження MCM та висновки комплексного типу
  - 5.4.2 Формальне визначення
    - 5.4.2.1 Властивості
    - 5.4.2.2 Обмеження конструкцій MCM
    - 5.4.2.3 інстанції дії правил
- 6. Правила
  - 6.1 Неформальний опис (ненормативний)
  - 6.2 Правила підтримки
  - 6.3 Правила, пов'язані з компонентами схеми
    - 6.3.1 Правила конструкцій MCM
    - 6.3.2 Схема компонентів правил
    - 6.3.3 Приклад правил переконливості
  - 6.4 Авторські правила в документах правил
    - 6.4.1 Обов'язкове правило
- 7. Локалізація природної мови повідомлень
  - 7.1 Підстановка змінних
- 8. Відповідність критеріям
- 9. Посилання Розширення MCM (ненормативні)
  - 9.1 Атрибути
    - 9.1.1 sml:acyclic - ациклічні
    - 9.1.2 sml:ref - посилальні
    - 9.1.3 sml:nilref – нульові посилання
    - 9.1.4 sml:targetElement – елементи мети
    - 9.1.5 sml:targetRequired
    - 9.1.6 sml:targetType – тип мети
    - 9.1.7 sml:locid
  - 9.2 Елементи
    - 9.2.1 Ключові елементи MCM, sml:key
    - 9.2.2 Ключові посилання MCM, sml:keyref
    - 9.2.3 Унікальні елементи MCM, sml:unique
    - 9.2.4 URI елементи MCM, sml:uri
  - 9.3 XPath функції
    - 9.3.1 smlfn: deref
- 10. Список літератури
  - 10.1 Нормативна
  - 10.2 Ненормативна

## Додатки

- A. Нормативна схема MCM
- B. Приклад визначення моделі документа (ненормативний)
- C. Приклад посилання MCM (ненормативний)
- D. Приклад URI посилання схеми MCM(ненормативний)
- E. Приклад обмеження ідентичності MCM (ненормативний)

- Ф. Зразки локалізації та підстановки змінних (ненормативний)
- Г. Подяки (ненормативний)

## 1. Вступ (ненормативний)

Мова Службового Моделювання (МСМ) забезпечує багатий набір конструкцій для створення моделей складних послуг і систем. Залежно від галузі застосування ці моделі можуть включати таку інформацію, як конфігурація, розгортання, моніторинг, політика, охорона здоров'я, планування, цільовий діапазон операційних угод про рівень обслуговування, і так далі. Моделі забезпечують значення такими важливими способами:

1. Зразковий фокус на захопленні всіх **інваріантних аспектів** сервіса/системи, який повинен підтримуватися сервісом/системою, щоб функціонувати належним чином.
2. Моделі являють собою могутній механізм **затвердження змін**, необхідний перед зверненням до сервісу/системи. А також, якщо зміни відбулися при запуску сервісу/системи, вони можуть заперечувати дані, описані в моделі.  
Фактичний сервіс/система і модель разом дозволяють сервісу/системі **самозагоювання** - остання задача. **Моделі сервісу/системи повинні обов'язково залишатися відокремленими від живого сервісу/системи, щоб створити контрольну петлю.**
3. Моделі - це одиниці **комунікації і співпраці** між конструкторами, виконавцями, операторами, і споживачами; і можуть легко розділятися, відстежуватись, і контрольню переглядатися. Це важливо, тому що комплексні послуги зазвичай будуються і підтримуються багатьма людьми, які відіграють різноманітні ролі.
4. Моделі призначені для **багатократного використання, і стандартизації**. Більшість світових комплексних сервісів і систем включають достатню кількість комплексних частин. Багатократне використання і стандартизація послуг/системи та їх складових - це ключовий коефіцієнт в зменшенні повної продукції і операційної вартості а також в збільшенні надійності.
5. Моделі дозволяють збільшити **автоматизацію** завдань управління. Засоби автоматизації, виставлені більшістю сервісів/систем, сьогодні могли б керуватися не людиною, а програмним забезпеченням, що призвело б як до надійної початкової реалізації сервісу/системи, так управління ongoing lifecycle.

Модель в МСМ реалізована як набір взаємозв'язаних документів XML-формату. Документи XML містять інформацію про складові сервісу, відомі як примушення, причому кожна частина повинна задовольнити вимогам даного сервісу, щоб функціонувати належним чином. Примушення задовольняються двома шляхами:

1. **Схема** - це обмеження за структурою і змістом документів у моделі. МСМ використовує XML схеми [\[XML схематичні структури, XML схематичні типи даних\]](#) в якості мови схем. Крім того МСМ визначає набір розширень для XML-схем задля підтримки посилань, які можуть перетинати кордони документа.

2. **Правила** - булеві вирази, які обмежують структуру та зміст документів в якості моделі. МСМ використовує схематрон [\[ISO / IEC 19757-3, "Вступ до схематрон, поліпшення перевірки схематроном\]](#).

Одна з важливих операцій моделі полягає у створенні її дії. Вона включає перевірку, тобто чи всі дані моделі задовольняють схеми і правила.

Ця специфікація зосереджується головним чином на визначенні розширеннями XML схем для посилань, які перетинають кордони документа, Схематрон використовується в МСМ, у процесі перевірки моделі. Передбачається, що читач знайомий з XML-схемами і схематроном.

Сценарії МСМ вимагають виконання кількох функцій, які або не виконуються, або не повністю підтримується в XML схемах. Ці функції можуть бути класифіковані в такий спосіб:

1. **МСМ посилання** - XML-документи представляють собою введення змісту через кордони, яке слід розглядати як єдине ціле. XML-схема не має підтримки для перехресних посилань, документів, хоча вона підтримує посилання на елементи в тому ж документі через `xs:ID`, `xs:IDREF`, `xs:key` і `xs:keyref`. Посилання між елементами визначені в окремих типових документів МСМ є суттєвими для специфікації МСМ. МСМ розширює XML схеми для підтримки посилань, які можуть перетинати кордони документа, а також набір обмежень на ці посилання, які застосовуються незалежно від того, чи перетнуть вони кордон документа, чи ні.

2. **Правила** – МСМ схема не підтримує мову для визначення довільних обмежень щодо структури та змісту документів МСМ. МСМ використовує схематрон, щоб виразити твердження про структуру та зміст документів XML.

XML схема підтримує дві форми розповсюдження: "атрибути у різних назвах" і "застосування інформаційних елементів"; обидві форми використовують розширення МСМ.

## 2. Позначення і термінологія

### 2.1 Умовні позначення

Ключові слова "НЕОБХІДНО", "НЕ ПОВИНЕН", "ПОТРІБНО", "НЕ", "НЕ", "СЛІД", "НЕ ПОВИННО", "РЕКОМЕНДУЄТЬСЯ", "МОЖЛИВО" ("MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" ) в цьому документі тлумачаться, як описано в RFC 2119 [\[IETF RFC 2119\]](#).

Дана специфікація використовує розширену форму позначення Бекуса-Наура (ABNF) [\[RFC 2234\]](#).

Ця специфікація слідує тієї ж конвенції для компонентів схем, які використовувалися в схемі специфікації XML [\[XML схематичні Структури\]](#). Тобто, посилання на властивості компонентів схем, таких як {приклад власності}, посилання на відповідні визначення, що зазначені у фігурних дужках. Посилання на властивості інформаційних елементів, як це визначено в [\[XML Information Set\]](#), такі як [діти], є нотними при посиланні на відповідний розділ, зазначені у квадратних дужках.

Ця специфікація посилається на такі терміни, як XML документ, елемент, атрибут і т. д. для стислості. Альтернативою цьому було б використовувати такі терміни, як "[XML документ або синтетичні Infoset](#)", "[елемент інформаційного елемента](#)", "[атрибут елемента інформації](#)" і т.д. Відповідно. Це зробило б специфікацію надмірно детальною без додавання або зміни значення існуючого тексту. Використання коротких термінів не виключає інших уявлень XML. Концепція, визначених у цій специфікації поширюються на всі форми уявлень XML.

Зміст цієї специфікації є нормативною, за винятком розділів або текстів, які явно позначені як ненормативні. Якщо розділи позначені як ненормативні, то всі вони містять підрозділи, що не є нормативними, навіть якщо вони чітко не позначені як такі. Усі записи не є нормативними, якщо не вказано інше.

## 2.2 Термінологія

Терміни, використані в даній специфікації перераховані в алфавітному порядку.

### **Документ**

Добре сформований XML документ, як визначено в [\[XML\]](#).

### **Визначається реалізацією**

Здійснення певних функцій або поведінки можуть відрізнятися у різних [моделей процесорів](#); точна поведінка незазначена у даній специфікації, але повинна бути визначена виконавцем кожної [моделі процесора](#).

### **Залежність від реалізації**

Залежність від реалізації функції або поведінки може відрізнятися у різних моделей процесорів; точна поведінка не передбачена цєю або будь-якою іншою специфікацією W3C, і не потрібна, при визначенні виконавцем для будь-якої конкретної реалізації.

### **Моделі**

Набір взаємопов'язаних [документів](#), що описують послуги або системи. Кожна [модель](#) складається з двох непересічних підмножин документів - модель визначення документів і типових документів інстанції.

### **Документи Зразкового Визначення**

Підмножина документів на моделі, яка описує схеми і правила, які регулюють структуру та зміст документів моделі. Ця специфікація визначає два види документа зразкового визначення: [схеми документів](#) і [правила документів](#), але дозволяє визначити реалізації інших видів документів, а також визначення моделі. Документи інших типів визначення

моделі не грають ніякої ролі в перевірці [достовірності моделей](#).

### **Документи Зразкового Прикладу**

Підмножина документів моделі, яка описує структуру та зміст модельованих осіб.

### **Моделі процесора**

Моделі процесора це втіленням того, що процеси відповідної моделі MCM використовуються в цілому або частково, семантика визначається цією специфікацією.

### **Перевірки моделі**

Перевірки моделі є процесом визначення, чи є та чи інша [модель](#) відповідною та дійсною. [8. [Відповідність критеріям](#)]

### **Модель валідатора**

Модель валідатора - це модель процесора, яка здатна виконувати перевірку моделі.

### **Правило**

Правило це логічне вираз, що стримує структура і зміст комплекту документів в моделі.

### **Зв'язуючі правила**

Зв'язуючі правила - це асоціація набору одного або декількох документів, \_правила набору нульових або більш типових документів. Документи, пов'язані з цим правилом документу, як кажуть, "пов'язані" з ним. Для моделі вважається дійсним [визначення кожного документу і екземпляру документу; в моделі](#) повинна відповідати обмеженням, які визначаються кожним правилом документа, до якого він прив'язаний. Це допустимо для правил документів, які не прив'язані пов'язані з ним, так і для моделі документа, яка буде пов'язана з нульовим правилом документу.

### **Правило документа**

Правило документа - це [документ визначений моделлю](#), що складається з обмежень схематрона.

### **Схема документа**

Схема документ - це документ визначений моделлю, який відповідає специфікації XML схем [\[XML схематичні Структури\]](#), визначений схемою документа.

### **Примушення схематрона**

Інформація, що міститься в одному `sch:schema` елементі.

### **Посилання MCM**

Посилання MCM - це елемент `sml:ref` атрибуту значенням якого є "правда".

#### **Примітка:**

Умоглядно, [довідка MCM](#) використовується, щоб сигналізувати зв'язок від одного елемента в моделі MCM до іншого елемента в тій же моделі.

### **Схема Посилання MCM**

Схема [Довідки MCM](#) - це набір правил, що визначають синтаксис, що використовується, щоб створити приклад довідкової схеми в контексті посилання MCM, плюс набір правил



для втілення прикладу довідкової схеми своєї мети. Всякий раз, коли "довідкова схема" відбувається в даній специфікації, слід припускати, що включає "схема довідки MCM" якщо не вказано інше. Незважаючи на схожі назви, згадка терміну схема MCM не має відношення до XPointer схем і URI схем.

### Мета

Елемент в моделі, вирішує [Посилання MCM](#) називається метою цієї довідки MCM.

### Повний ідентифікатор мети

Цільовий повний ідентифікатор URI або IPI, який містить всю інформацію, необхідну для виявлення даного посиланням MCM. Саме наслідком цього визначення є те, що повний ідентифікатор [мети](#) не може бути відносний URI/IRI.

## 2.3 Простори імен MCM

Таблиця 2-1 списків імен XML, які використовуються в даній специфікації. Вибір будь-якого префікса для простору імен є довільним і семантичний істотно..

Таблиця 2-1. Простори імен XML, що використовуються в даній специфікації.

Префікс	XML Namespace	Специфікація
sml	<a href="http://www.w3.org/ns/sml">http://www.w3.org/ns/sml</a>	Дана специфікація
smlfn	<a href="http://www.w3.org/ns/sml-function">http://www.w3.org/ns/sml-function</a>	Дана специфікація
xs	<a href="http://www.w3.org/2001/XMLSchema">http://www.w3.org/2001/XMLSchema</a>	<a href="#">[Структури XML Схема, XML схема типів даних]</a>
sch	<a href="http://purl.oclc.org/dsdl/schematron">http://purl.oclc.org/dsdl/schematron</a>	<a href="#">[ISO/IEC 19757-3]</a>

## 3. Залежність від інших специфікацій

Інші умови, від яких залежить перераховані в [\[Нормативно-посилання\]](#).

## 4. Посилання MCM

Підтримка для [посилання MCM](#) в [моделі](#) MCM включає в себе:

1. Можливість використання декількох посиланням схем MCM посиланням MCM.
2. Розширення механізму, що дозволяє новим схемам довідки MCM визначатися.
3. Обмеження на тип посилання елемента.
4. Можливість визначити ключових, унікальних і основних орієнтирів обмежень за посиланням MCM.

[Моделі валідатори](#) повинні підтримувати [посилання MCM](#) так, як вони визначені в даній специфікації.

[Додаток С](#) Приклад посилання MCM - це приклад, який показує, як посилання MCM визначаються і використовуються.



## 4.1 Визначення посилання MCM

### 4.1.1 Посилання MCM

Пункт елементів інформації у випадку, коли модель документа MCM є посиланням MCM, якщо і тільки якщо вона має атрибут, інформація якої з впливає з наступних умов:

1. Його [\[місцева назва\]](#) є посиланням
2. Його [\[простір імен\]](#) є <http://www.w3.org/ns/sml>
3. Його [\[нормовані значення\]](#) після пробілу нормалізації використовують `collapse` наступних схематичних правил це або `"true"` або `"1"`.

#### Примітка:

Цей механізм дозволяє меншій схемі ідентифікувати посилань MCM, тобто посилання MCM можуть бути визначені, не покладаючись від Поштового Схематичного Твердження Інфосет (PSVI). [\[XML схема Структури\]](#)

Вона визначається реалізацією, коли моделі процесорів, які не є одночасно моделями використання валідаторів MCM Інфосет [\[XML Information Set\]](#) або поштової перевірки схем Інфосет (PSVI) [\[XML схема Структури\]](#) для ідентифікації посиланням MCM.

#### Примітка:

Валідатори моделі MCM слід використовувати, щоб визначити PSVI посилання MCM. Див [8. Відповідність критеріям](#).

Посилання MCM вважається примірником конкретного посилання схеми MCM, якщо вона може бути ідентифікована як така у відповідності з правилами цієї схеми MCM . Див [4,3 MCM опорних схем](#). MCM посилання може бути примірником декількох опорних схем MCM.

Хоча її нормативне визначення дозволяється відразу декільком синтаксисам, які будуть використовуватися для виявлення посиланням MCM для стислості і послідовність, на решті частині цієї специфікації використовується `sml:ref="true"` ,щоб позначити посилання MCM в прикладах і текстах.

Наступний приклад ілюструє посилання MCM на прикладі URI опорних схем MCM.

```
<RefElement sml:ref="true">
  <sml:uri>targetDocument.xml</sml:uri>
</RefElement>
```

### 4.1.2 Недійсні посилання MCM

Посилання MCM є нульовим, якщо і тільки якщо вона має атрибут, інформація для якої в наступному справджується.

1. Його [\[місцева назва\]](#) є `nilref`
2. Його [\[простір імен\]](#) є `http://www.w3.org/ns/sml`
3. Його [\[нормовані значення\]](#) після пробілу нормалізації використовують `collapse` наступних схематичних правил це або `"true"` або `"1"`.

Це є наслідком вищесказаного, тобто ця специфікація не присвоює значення `sml:nilref` атрибутів, коли вона використовується на елемент, який не є посилання MCM. Моделі валідаторів можуть обиратися, щоб попередити про виявити цієї умови в документі.

Наступний приклад ілюструє нульове посилання MCM.

```
<RefElement sml:ref="true" sml:nilref="true">  
  <sml:uri>targetDocument.xml</sml:uri>  
</RefElement>
```

#### Примітка:

SML: `nilref` може бути корисним у випадку, коли автор визначає схему комплексного типу за значенням SML: REF = "True" з фіксованим значенням "true", але примірник автору сигналізує про відсутність мети.

Вона визначається реалізацією чи моделлю процесорів, які не є одночасно моделями використання валідаторів MCM Інфосет [\[XML Information Set\]](#) або поштової перевірки схем Інфосет (PSVI) [\[XML схема Структури\]](#) для визначення нульового посилання MCM.

#### Примітка:

Валідаторів моделі MCM повинні використовувати PSVI, щоб визначити недійсні дідки MCM. Див [8. Відповідність критеріям](#).

### 4.1.3 Невирішені посилання MCM

Посилання MCM є невіршеними, тоді і тільки тоді, коли справджується наступне::

1. Це ненульове посилання MCM.
2. Жодна з довідкових схем, яку посилання MCM визнало використаною, не встановлює елементи у моделі.

#### Примітка:

Поняття невіршеного посилання контекстно-залежне. Тобто, різні процесори моделі, заснованої на безлічі опорних схем MCM, які вони розуміють і використовують в моделюванні даного процесу, можуть розглядати різні посилання MCM невіршеними.

Наступний приклад ілюструє, невіршені посилання MCM (припускаючи, що цей документ `dummy.xml` не існує в моделі).

```
<RefElement sml:ref="true">  
  <sml:uri>dummy.xml</sml:uri>
```

```
</RefElement>
```

#### 4.1.4 Мета посилання MCM

Елемент вузла, який являється недійсним посиланням MCM, і вирішується є його метою. Мета посиланням MCM повинна бути частиною однієї і тієї ж моделі MCM, як вихідна MCM. Недійсні довідки посилання MCM не мають мети.

Метод визначення того, які документи є частиною моделі MCM визначається реалізацією.

##### Примітка:

Наприклад, модель MCM може складатися з документів, перерахованих у файлі конфігурації, або модель MCM може розглядатися, як перехідне закриття документів, зазначених у будь-якому посиланні MCM, починаючи з набору документів відомих в даній моделі.

Наступний приклад ілюструє посилання MCM, який має на меті другий елемент дитина - кореневий елемент документа target.xml.

```
<RefElement sml:ref="true" xmlns:e="urn:example">
  <sml:uri>target.xml#smlxpath1(e:Course[2])</sml:uri>
</RefElement>

document 'target.xml':
-----
<Courses xmlns="urn:example">
  <Course>
    <Name>PHY101</Name>
    <Grade>A</Grade>
  </Course>
  <Course>
    <Name>MAT101</Name>
    <Grade>A</Grade>
  </Course>
</Courses>
```

## 4.2 Семантика посилання MCM

Моделі валідатора повинні спробувати вирішити посиланням MCM з використанням усіх опорних схем, яких посиланням SML є визнаним як приклад.

### 4.2.1 Не більше однієї мети

Кожне недійсне посиланням MCM повинно бути орієнтовано на якнайбільше один елемент моделі. Це означає, що кожна визнана довідкова схема, що використовується в довідці MCM, не ПОВИННА вирішувати до більш ніж одну мету.

Наступний приклад ілюструє довідку MCM, яка порушує правило більш ніж однієї мети:

```
<RefElement sml:ref="true" xmlns:e="urn:example">
  <sml:uri>target.xml#smlxpath1(e:Course)</sml:uri>
</RefElement>

document 'target.xml':
-----
<Courses xmlns="urn:example">
  <Course>
    <Name>PHY101</Name>
    <Grade>A</Grade>
  </Course>
  <Course>
    <Name>MAT101</Name>
    <Grade>A</Grade>
  </Course>
</Courses>
```

#### 4.2.2 Послідовні посилання

Якщо недійсне посилання МСМ є екземпляром багаторазових довідкових схем, всі визнані довідкові схеми ПОВИННІ вирішувати одну і ту саму мету, або всі вони повинні бути невирішеними.

#### 4.2.3 Ідентична мета

Щоб визначити, що дві мети являються однаковими або навпаки - різні, модельні валідатори повинні підкорятися наступним правилам:

1. Якщо обидві процедури відповідають дійсності, то моделі валідаторів повиненні враховувати і однаковість мети обох процедур.

А. Визначення вихідної схеми (S) конкретизує, яким чином URI перетворюються на повні ідентифікатори мети.

В. Два повні ідентифікатори мети ідентифікуються використовуючи реєстр, Codepoint-By-Codepoint порівняння.

2. В іншому випадку, модель валідатора повиненна враховувати обидві мети, щоб відрізнитися, коли є щось доступне в елементі інформації, яка розрізняє їх. Наприклад, якщо є властивість інфосету, для якої дві мети мають різне значення, вони являються різними. Це рекурсивно відноситься до комплекснозначних властивостей.

3. Для всіх інших випадків це визначається реалізацією, при умові, що мета є такою ж або ні.

#### 4.2.4 Множинні посилання

Елемент в документі може бути націлений на численні посилання МСМ.

#### 4.2.5 Недійсні посилання МСМ

Недійсні посиланням МСМ - це явна декларація намірів автора документа, мета данного

посилання МСМ не існує. Якщо посилання МСМ визнане недійсним, то модель процесорів не повинна намагатися визнавати будь-які посилання схем, що використовуються в ньому.

#### 4.2.6 Детермінована оцінка обмежень МСМ

Кожне недійсне посиланням МСМ повинно задовольняти всі наступні умови, для того, щоб мати можливість детерміновано оцінити обмеження МСМ і правила, пов'язані з нею.

1. Посилання повинно мати не більше однієї мети. [\[4.2.1 не більше однієї мети\]](#)
2. Посилання має бути послідовним. [\[4.2.2 Послідовні довідки\]](#)

#### 4.2.7 smlfn: deref () - функція розширення XPath

Функція Deref () приймає набір вузлів елементів і повертає набір вузлів, що складається з вузлів елементів, відповідних елементів, на які посилається вхідний набір вузлів. Зокрема, для кожного R посилання МСМ на вхідному вузлі встановлюється набір вихідних вузлів, що містить більше одного елемента вузла.

Нехай I = вхідний набір вузлів, то є безліч вузлів, що передаються до deref () функції.

Нехай O = вихідний набір вузлів, то є безліч вузлів, що повертається до deref () функції.

Поведінка deref () функції повинна задовольняти наступні обмеження:

1. Для кожного R посилання МСМ на вхідному вузлі ставимо I:

A. Якщо реалізація не визнає ніякої схема МСМ, що використовується в R посиланні МСМ, тоді ні один елемент не приєднаний до O.

B. Якщо реалізація визнається R, наприклад N підтримує опорні схеми, то deref () не потрібна для вирішення всіх N схем. Його поведінка в цьому випадку визначається реалізацією і набором опорних схем, які фактично можуть бути будь-якою підмножиною зазначених схем. Це залежить від наступних чинників:

I. Якщо deref () не намагається вирішити будь-яку довідкову схему або посилання, то ні один елемент не додається до O.

II. Якщо як мінімум одне з рішень довідкових схем перетвориться в більш ніж одну мету елементу, то 0 або 1 з цілей додається до O.

III. Якщо одна довідкова схема, що використовується вирішує мету, відмінну від мети, яка вирішується іншою довідковою схемою, то 0 або 1 з цілей додається до O.

IV. Якщо одна довідкова схема усуває, а інша ні, то 0 або 1 з цілей додається до O

V. Якщо жодне з перерахованих вище тверджень, не відповідає дійсності (тобто, всі спроби довідкових схем зводяться до однієї і тільки однієї мети елемента, позначимо T), то один елемент мети (а саме, T) додається до висновку, якщо він ще не існує в O.

#### Примітка:

Вище сказане описує поведінку, необхідну для загальної XPath 1.0 `deref ()` функції, в цій якості експонат має кілька суттєвих відмінностей від поведінки, яка необхідна для перевірки посилання MCM під час перевірки моделі. По-перше, вона може бути використана для успішного процесу інстанції документу, в якому модель MCM невідома або недійсна, хоча результати в цьому випадку не можуть бути сумісні. По-друге, відколи XPath 1.0 не проходить від функції до помилкового сигналу, поведінка задана для повернення результатів до довідок MCM, які не підкоряються виконанню всіх правил, наприклад, посилання в якому XPath вираз використовується більш, ніж в одному вузлі. Як йдеться в цьому розділі, такої функції буде недостатньо, щоб перевірити правильність посилання MCM.

Моделі валідатори повинні забезпечити здійснення `deref ()` XPath функції розширення. На додаток до вищевказаних вимог, для загальної реалізації функції `deref ()`, для кожного посилання MCM використовується визнана схема, `deref ()` в моделі валідатора повинна спробувати вирішити принаймні одну з визнаних схем.

## 4,3 Схеми посилань MCM

Посилання MCM може бути прикладом цілого ряду опорних схем. MCM не пропонує використання будь-яких конкретних схем введення. Схеми введення MCM можуть використовувати дочірні елементи, атрибути, обидва, або жоден для збору відомостей, необхідних для ідентифікації введеної мети. Необов'язково, щоб всі елементи в моделі MCM були доступними посиланню MCM. Це буде залежати від того, чи підтримує це дана опорна схема.

Хоча MCM не вимагає використання будь-якої конкретної схеми, це вказує, як посилання повинно бути представлене при використанні певних опорних схем MCM. Ця специфікація визначає [4.3.1 Опорні схеми URI](#) для використання в посиланнях MCM.

Посилання опорної схеми MCM повинно відповідати всім наступним характеристикам:

1. Набір правил, при виконанні яких, посилання MCM визначається як приклад схеми. Посилання опорної схеми може накласти додаткові вимоги на дії посилань MCM, що визнаються екземплярами цієї схеми. Моделі валідаторів НЕ МОЖУТЬ застосовувати такі вимоги до посилання MCM, які не є екземплярами відповідних опорних схем.
2. Набір правил, які задовольняються, при вирішенні посилання MCM на даному вузлі елементу мети.
3. Твердження, що держава, як випадок опорних схем, перетворюється на повні ідентифікатори мети. Якщо вони перетворюються на повні

ідентифікатори мети, посилання опорної схеми повинне описувати процес трансформації.

Посилання опорної схеми повинне вказувати всі попередні пункти, які застосовуються до дійсних випадків опорної схеми MCM, і може вказувати їх для інших (недійсних) випадків.

#### 4.3.1 URI посилання схеми MCM

Опорна URI схема MCM визначається наступним чином:

1. Посилання MCM визначається як екземпляр опорної URI схеми MCM, якщо і тільки якщо рівно один елемент інформаційного елемента [\[XML Information Set\]](#) якого [\[місцева назва\]](#) є URI, і чиї [\[простор імен\]](#) є <http://www.w3.org/ns/sml> присутня як дитина цього опорного елемента.

Наприклад опорна схема MCM є дійсною, якщо вона відповідає всім наступним вимогам.

A. Зміст URI елемента повинен бути типу XS: anyURI як це визначено в специфікації схем XML [\[XML схема типів даних\]](#).

B. Ідентифікатор фрагмента (якщо такий наявний) повинен відповідати синтаксису однієї з наведених нижче процедур.

##### I. 4.3.1.1 smlxpath1 () схема

##### II. [Стенографічний показчик](#)

2. Посилання MCM, яке є екземпляром опорної URI схеми MCM буде вирішена за допомогою наступних кроків:

A. Документ XML **D** отримують у такий спосіб:

I. Якщо посилання URI є посиланням данного ж документа, визначеного у відповідному URI RFC, то **D** є документом, що містить посилання MCM.

II. В іншому випадку, **D** визначається наступним чином:

A. Якщо посилання URI є відносним посиланням, тоді нехай **U** - результат регулювання посилання, використовуючи [ базові URI] властивості [\[XML набору інформації\]](#) `<sml:uri>` елемента, як базового URI. В іншому випадку, **U** - це власне посилання URI. Розрахунок [базової URI] властивості визначається реалізацією.

B. **U** як визначення у відповідних специфікаціях. Якщо документ є ціллюю **U** і знаходиться в поточній моделі MCM, то **D** являється даним документом. В іншому випадку, якщо



документ не знаходиться в поточній моделі MCM, то приклад опорної URI схеми MCM є невіршеним (і **D** не має значення).

**Примітка:**

У результаті наведеного вище визначення, якщо отриманий об'єкт не є медіа типом XML, або якщо він не дуже добре сформований XML, то, за визначенням, цей об'єкт не є документом, як це визначено в даній специфікації. У цьому випадку, приклад опорної схеми MCM є невіршеним.

В. Якщо фрагмент компонента присутній на посиланні URI, то, наприклад, опорна URI схема MCM усуває кореневий елемент **D**.

С. Якщо фрагмент компонента присутній на посиланні URI, то відповідна обставина використовує наступне:

I. Якщо фрагмент компонента відповідає синтаксису `smlxpath1 ()` XPointer схеми, то посилання має на меті отримати фрагмента компонента **D** шляхом застосування, як це визначено в розділі [4.3.1.1 smlxpath1 \(\) схема](#).

II. Якщо фрагмент компонента відповідає синтаксису покажчика стенографії, то відповідна обставина використовує наступне:

A. Якщо мета **T** може бути виявлена в ході роботи на основі схеми виявлення ID MCM, то метою посилання є **T**.

B. Якщо мета в ході роботи не може бути визначена на основі схеми виявлення ID MCM, то вона визначається реалізацією, мається на увазі, мета посилання в **D** визначається на основі інших критеріїв дозволених покажчиком стенографії.

3. Випадки, коли опорна URI схема MCM перетворюються на повний ідентифікатор мети через стандартну URI обробку, описані у відповідному URI RFC.

Наступний приклад ілюструє посилання MCM, яке є прикладом опорної URI схеми MCM. Посилання має на меті елемент ID `targetId` в документі `target.xml`.

```
<RefElement sml:ref="true">
  <sml:uri>target.xml#targetId</sml:uri>
</RefElement>
```

#### 4.3.1.1 `smlxpath1 ()` схема

`Smlxpath1 ()` схема призначена для використання Рамкового покажчика [XPointer], щоб дозволити вирішення елементів. 4.3.1 Опорна URI схема MCM використовує її для кодування ідентифікаторів фрагменту.

У цьому розділі описуються синтаксис і семантика `smlxpath1 ()` схеми і поведінка

процесора XPath по відношенню до цієї схеми.

1. Ім'я схеми: smlxpath1
2. Синтаксис схеми використовує ABNF [\[RFC 2234\]](#):

```
SMLXPath1_Fragment_ID ::= 'smlxpath1' '(' SMLXPath1_SchemeData ')'
```

```
SMLXPath1_SchemeData ::= XPath1.0_LocationPath
```

де,

XPath1.0\_LocationPath є виробництвом LocationPath, яке визначене в 1,0 специфікації XPath [\[XPath\]](#).

3. Розширена функція Deref () XPath не повинна бути зображеною в контексті вираження бібліотечної функції при обробці шляху до SMLXPath1\_SchemeData.
4. Простір імен Зв'язаного Контексту: smlxpath1 () схема успадковує безліч іменних прив'язок, які є в розпорядженні носіїв sml:uri елементів.
5. Для даного документа **D**, елемент, націлений схемним прикладом, одержується шляхом застосування шляху до кореневого елемента **D** SMLXPath1\_SchemeData . Результат повинен бути або 1 вузлом елемента, або бути порожнім. В іншому випадку, результат XPath є помилковим.

#### Примітка:

В інших випадках опорної URI схеми MCM, **D** є документом, вирішений по нефрагментній частині посиланням URI, як він визначений в пункті 2.а в розділі [4.3.1 URI посилання схеми MCM](#).

Наступний приклад ілюструє посилання MCM, яке є прикладом опорної URI схеми MCM. Посилання має на меті кореневий елемент документа `target.xml`.

```
<RefElement sml:ref="true">  
  <sml:uri>target.xml#smlxpath1(/*)</sml:uri>  
</RefElement>
```

## 5. Обмеження MCM

### 5.1 Обмеження на посилання MCM

MCM підтримує наступні атрибути для вираження обмеження на посилання MCM.

Таблиця 5-1. Атрибути

Назва	Опис
MCM: ациклічність	Використовується, для вказівки, що цикли заборонені для посиланням MCM.

МСМ: необхідна мета	Використовується, для зазначення того цільового елемента посиланням МСМ, який повинен бути присутнім в моделі.
МСМ: елемент мети	Використовуються для обмеження назви мети посилання МСМ.
МСМ: тип мети	Використовуються для обмеження типів мети посилання МСМ.

МСМ визначає нову властивість для кожного комплексу типу визначення компонента схеми:

### **{Ациклічність}**

Логічне значення `xs:boolean`. Необхідне.

Значення {Ациклічності} для `xs:anyType` є помилковим.

МСМ визначає три нових властивості для кожної складової елементу декларації:

### **{Необхідна мета}**

Логічне значення `xs:boolean`. Необхідне.

### **{Елемент мети}**

Складовий елемент декларації. Не обов'язковий.

### **{Тип мети}**

Визначення типу компонента. Не обов'язкове.

## **5.1.1 Ациклічність МСМ (`sml:acyclic`)**

`sml:acyclic` використовується, щоб вказати, чи слід допускати цикл, чи ні у випадку складного типу. Моделі валідатора повинні підтримувати `sml:acyclic` атрибути будь-якого `<xs:complexType>` елемента в схемі документа. Цей атрибут типу `xs:boolean` і його фактичне значення може бути або істинними або помилковими.

### **5.1.1.1 Обмежені конструкції МСМ**

{Ациклічні} властивості значення комплексних типів визначення є такими, як це зазначено у відповідному випадку серед наступних:

1. Якщо `sml:acyclic` присутній, то {ациклічність} має фактичне значення цього атрибута.
2. В іншому випадку, якщо його {базовий тип} визначення являє собою комплексний тип визначення, а потім {ациклічні} мають те ж саме значення {ациклічності}, як її {базовий тип визначення}.
3. В іншому випадку ({базовий тип визначення } є простим визначенням типу),

{ациклічність} є помилкою.

### 5.1.1.2 Схема компонентів правил

Якщо комплекс типу визначення **СТ** (база типу визначення) є також комплексом типу визначення і має справдження {ациклічності}, то **СТ** ПОВИНЕН справджуватись для {ациклічності}.

### 5.1.1.3 Інстанції дії правил

Якщо **СТ** є комплексом типу визначення з {ациклічності}, то **СТ** НЕ МАЄ створювати цикли в моделі. Точніше, орієнтована діаграма, яка будується наступним чином, повинна бути ациклічною:

1. Вузли у діаграмі являються всіма елементами, що вирішуються посиланням MCM типу **СТ** або типів, які є похідними від **СТ**.
2. Якщо вузол **N** в діаграмі містить **R** посилання MCM типу **СТ** або типу, отриманого з **СТ**, і **R** вирішує **T** (який також повинен бути вузлом у діаграмі), то дуга взята з **N** до **T**.

### 5.1.2 Обмеження на меті посилання MCM

MCM визначає три атрибути: `sml:targetRequired`, `sml:targetElement` і `sml:targetType`, для стримування мети посилання MCM. Ці три атрибути разом називаються `sml:target` атрибутами. Моделі валідатора повинні підтримувати ці атрибути в усіх `xs:element` елементах з ім'ям атрибута. Обмеження `sml:target*` кріпляться до елемента декларації компонента схеми.

#### 5.1.2.1 Обмеження конструкцій MCM

1. [{Необхідність мети}](#), зазначені у відповідному випадку серед наступних:

- A. Якщо `sml:targetRequired` присутня, то [{Необхідність мети}](#), є фактичним значення цього атрибута.
- B. В іншому випадку, якщо елемент декларації має {приналежність до групи}, то [{Необхідність мети}](#) є тою ж самою {приналежністю до групи}
- C. В іншому випадку [{Необхідність мети}](#) є помилковою.

2. [{Необхідність мети}](#), зазначені у відповідному випадку серед наступних:

- A. Якщо `sml:targetElement` присутній, то її фактичне значення має вирішувати глобальний елемент декларації **G** і [{Необхідність мети}](#) є **T**.
- B. В іншому випадку, якщо {приналежність до групи} присутня, то [{необхідність мети}](#), } є тою ж самою {приналежністю до групи}
- C. В іншому випадку [{необхідність мети}](#) відсутня.

3. {Необхідність мети}, зазначені у відповідному випадку серед наступних:
- А. Якщо `sml:targetType` присутній, то цього фактичне значення має вирішувати глобальний тип визначення **T** і [{необхідність мети}](#) є **T**.
  - В. В іншому випадку, якщо {приналежність до групи} присутня, то [{необхідність мети}](#) є тою ж самою {приналежністю до групи}
  - С. В іншому [випадку {необхідність мети}](#) відсутня.

### 5.1.2.2 Схема компонентів правил

Моделі валідатора повинні втілювати наступне:

1. Якщо глобальний елемент декларації **E** має {приналежністю до групи} **G**, то значення мети **E** МСМ обмежує властивість **P** (одну з {необхідності мети},{елемент мети} або {тип мети}), повинні бути відповідним чинним обмеженням властивостей **G**, як це визначено в розділі [5.3 Діючі обмеження значень обмеження МСМ](#).
2. Якщо два елемента декларації **E1** і **E2** мають один і той же ж {простір імен} та {назву} і обидва вони містять (прямо, опосередковано або неявно) у змісті модель комплексного типу, то **E1** і **E2** мають однакові {необхідності мети},{елемент мети} і {тип мети}.

#### Примітка:

Дана умова № 2 на використання `sml:target*` атрибутів була визначена для зменшення навантаження на реалізацію моделі валідатора. Будь ласка, зверніться до розділу [5.4.1 Огляд обмеження обробки МСМ та висновок комплексного типу](#) для отримання додаткової інформації.

### 5.1.2.3 Інстанція дійсності правил

Якщо елемент декларації **E** має [{необхідності мети}](#), що являється дійсністю, то кожен приклад елемента **E**, який є також посиланням МСМ, повинен бути орієнтованим на деякі елементи моделі. Тобто, немає приклада **E**, що може бути нульовим або недозволеним посиланням МСМ.

Якщо елемент декларації **E** має [{елемент мети}](#) **TE**, то кожен приклад елемента **E**, який також є вирішенням посилання МСМ, повинен бути орієнтованим на елемент, який є прикладом **TE** чи прикладом деякого глобального елемента декларації в об'єднанні групи **TE**.

Якщо елемент декларації **E** має [{тип мети}](#) **TT**, то кожен приклад елемента **E**, який є також посиланням МСМ, повинен бути орієнтованим на елемент, [тип визначення] якого є **TT** або тип, отриманий з **TT**.

### 5.1.3 Короткий виклад обмежень посилань МСМ (ненормативний)

Результат вищезгаданого, як приклад перевірного правила, наводяться в таблиці нижче.

Таблиця 5-2. Цільові обмеження та категорії посилань MCM.

Номер категорії.	Ациклічні	Необхідна мета	Елемент мети	Тип мети
Номер посилання	Задовільняється	Задовільняється	Задовільняється	Задовільняється
Недійсні	Задовільняється	Порушується	Задовільняється	Задовільняється
Нерозпізнані	Задовільняється	Порушується		Задовільняється
Стійкі	Перевіряється	Задовільняється	Перевіряється	Перевіряється

"Перевірити" в таблиці вищесказане означає, що відповідні обмеження повинні бути оцінені.

Обмеження описані вище, можуть бути корисні навіть на елементі заяви, екземпляри якої є необов'язково посиланням MCM, оскільки рішення про те, чи включати обмеження і рішення про те, чи слід зробити елемент посиланням MCM, що можна зробити самостійно - деякі рішення приймаються схемою автора, інші виконані за прикладом документа автора.

## 5,2 Ідентичність обмежень MCM

Схема MCM підтримує визначення унікальності та обмеження посилання через `xs:key`, `xs:unique`, and `xs:keyref` елементів. Однак масштаби цих обмежень обмежені єдиним документом. MCM визначає аналоги цих обмежень, масштаби яких поширюється на численні документи, дозволивши їм пройти через посилання MCM.

Моделі валідатора повинні підтримувати наступні елементи для визначення ідентичності обмежень MCM через посилання MCM, як дочірні елементи `xs:element/xs:annotation/xs:appinfo`, де `xs:element` має назву атрибута.

Назва	Опис
ключова MCM	схожі на <code>xs:key</code> , за винятком, коли селектор і область XPath виразу можуть використовувати <code>smlfn:deref</code> функцію
унікальна MCM	схожі на <code>xs:unique</code> за винятком, коли селектор і область XPath виразу можуть використовувати <code>smlfn:deref</code> функцію
ключові посилання MCM	схожі на <code>xs:key</code> , за винятком, коли селектор і область XPath виразу можуть використовувати <code>smlfn:deref</code> функцію

Додаток В. [Приклад визначення моделі документа](#) і Додаток Е. [Приклад обмеження ідентичності MCM](#) включають приклади, які показують, як визначаються обмеження ідентичності MCM.

Обмеження ідентичності MCM закріплені до схематичного компонента декларації елементу. MCM визначає нові властивості для кожного схематичного компонента декларації елементу:

## {Визначення обмежень ідентичності MCM}

Набір компонентів визначень обмежень ідентичності MCM, які мають однаковий набір властивостей, як XML схема визначення обмежень ідентичності.

### 5.2.1 Синтаксис і семантика

Назви всіх визначень обмежень ідентичності існують в єдиному просторі символів, який не перетинається з будь-якими символами простору компонентів XML Схеми.

#### 5.2.1.1 Конструкції обмежень MCM

Для кожного елемента `sml:key`, `sml:unique`, або `sml:keyref` без `ref` атрибуту зазначено, що {Визначення обмежень ідентичності MCM} містить компонент, відповідний цьому елементу, як зазначено в пункті 3.11 визначення обмежень ідентичності специфікації XML Схеми [XML Структура схеми], де `sml:selector` та `sml:field` елементи використовуються в `xs:selector` та `xs:field`.

Для кожного `sml:key`, `sml:unique`, або `sml:keyref` елемента з `ref` атрибутом зазначено, {Визначення обмежень ідентичності MCM} містить компонент, вирішений на фактичним значенням `ref` атрибуту, з дотриманням наступних умов:

1. Назва атрибуту НЕ ПОВИННА бути вказаною.
2. Кореневі елементи `sml:selector` та `sml:field` не повинні бути вказані.
3. Якщо елемент є `sml:key`, то значення `ref` атрибуту повинно вирішувати ключове обмеження MCM.
4. Якщо елемент є `sml:unique`, то значення `ref` атрибуту повинно вирішувати унікальні обмеження MCM.
5. Якщо елемента є `sml:keyref`, то значення `ref` атрибуту повинно вирішувати ключові обмеження посилань і `refer` атрибут не повинен бути вказаним.

На додаток, обмеження ідентичності MCM отримуються з наведеного вище визначення або посилання, якщо елемент декларація **S** має {приналежністю до групи} **G**, то його {Визначення обмежень ідентичності MCM} містить також члени {Визначення обмежень ідентичності MCM} **G**.

#### 5.2.1.2 Правила компонентів схеми

1. XPath вираз `sml:selector` має той же синтаксис, який визначений в селекторі XML схеми обмеження ідентичності, як XPath синтаксис з одним виключенням. `sml:selector` XPath МОЖЕ використовувати `smlfn:deref()` функцію, з функціональним викликом викладенх на будь-яку глибину, на початку цього виразу. На підтримку цієї вимоги Пас виробництво внесло поправки щодо XML Схеми обмеження ідентичності, як зазначено нижче.

```
1. Path ::= ('//')? Step ( '/' Step)* | DerefExpr
```



```
2.   DerefExpr ::= NCName ':' 'deref' '(' Step ('/'Step)* ')' ('/'Step)*
3.   NCName   ':' 'deref' '(' DerefExpr ')' ('/'Step)*
```

2. XPath вираз `sml:field` має той же синтаксис, який визначений в селекторі XML схеми обмеження ідентичності, як XPath синтаксис з одним виключенням. `sml:field` XPath МОЖЕ використовувати `smlfn:deref()` функцію, з функціональним викликом викладенх на будь-яку глибину, на початку цього виразу. На підтримку цієї вимоги Пас виробництво внесло поправки щодо XML Схеми обмеження ідентичності, як зазначено нижче.

```
1. 4 Path ::= ('.//')? ( Step '/' ) * ( Step | '@' NameTest ) |
2. DerefExpr ('/' '@' NameTest)?
3. DerefExpr ::= NCName ':' 'deref' '(' Step ('/'Step)* ')' ('/'Step)* |
4. NCName ':' 'deref' '(' DerefExpr ')' ('/'Step)*
```

3. {Визначення обмежень ідентичності MCM} з елементів декларації НЕ МАЄ містити два обмеження ідентичності з однією і тією ж назвою.

#### Примітка:

Це може статися у випадку, якщо атрибут посилання вирішує обмеження ідентичності, які вже містяться в тому ж елементі декларації {Визначення обмежень ідентичності MCM}.

1. Якщо глобальний елемент декларація **E** має {приналежністю до групи} **G**, то значення **E** властивостей {Визначення обмежень ідентичності MCM} ПОВИННА бути дійсним обмеженням значення відповідної властивості **G**, як це визначено в розділі 5.3 Діє Чинні Обмеження значення Обмеження MCM.

2. Якщо два елементи декларацій **E1** і **E2** мають однаковий простір імен і назву, обидва містяться (прямо, опосередковано або неявно) в моделі вмісту складного типу, то **E1** і **E2** повинні мати однаковий набір {Визначення обмежень ідентичності MCM}

#### Примітка:

Це правило визначено для зменшення навантаження на реалізацію моделі валідатора. Будь ласка, зверніться до розділу 5.4.1 Огляд обробки обмеження MCM та висновок комплексного типу для отримання додаткової інформації.

### 5.2.1.3 Інстанції дії правил

Правила перевірки для обмежень ідентичності MCM такі ж самі, як зазначено в пункті 3.11 [Визначення обмежень ідентичності](#) для ідентичності XML Схем [XML Структура схеми]), з додаванням підтримки `smlfn:deref()` функції.

## 5.3 Чинні Обмеження значень обмежень MCM

Нехай  $BV$  = Значення властивості обмеження **P** (одне з [необхідності мети](#), [елемент мети](#)) і [тип мети](#)).

Нехай  $RV$  = значення, яке обмежує  $BV$ .

Для  $RV$ , щоб бути дійсними обмеження  $BV$ , відповідний приклад серед наступних має справджуватися:

1. Для [{необхідності мети}](#) серед наступних відповідний приклад застосовується.
  - A. Якщо  $BV$  справджується, то  $RV$  також справджується.
  - B. Якщо  $BV$  є помилковим,  $RV$  являється або істиною, або помилкою.
2. Для [{елементу мети}](#) застосовується наступне:
  - A.  $RV$  таке ж як  $BV$ .
  - B.  $RV$  в заміні групи  $BV$ .
3. Для [{типу мети}](#) застосовується наступне:
  - A.  $RV$  таке ж як  $BV$ .
  - B.  $RV$  – це тип, отриманий з  $BV$ .
4. Для [{Визначення обмеження ідентичності MCM}](#) одним з таких застосовується наступне:
  - A. Якщо  $RV$  таке ж як  $BV$ , то все наступне справджується:
    - I Число записів в  $RV$  таке ж, як і кількість записів у  $BV$ .
    - II. Для кожного запису в  $BV$ , існує запис у  $RV$  з тією ж кваліфікованою назвою (`{name}` + `{target namespace}`).
  - B.  $RV$  є надбудовою  $BV$ . Тобто,  $RV$  має всі записи з  $BV$ , так як це зазначено у попередньому пункті, і має один або декілька додаткових записів.

## 5,4 Обмеження MCM і висновок комплексного типу

### 5.4.1 Огляд обробки обмеження MCM та висновок комплексного типу

Цей розділ ненормативний.

Для комплексного типу  $D$ , отриманого від його (визначення базового типу)  $B$ , якщо  $D$  включає елемент декларації  $ED$  і елемент декларації  $EB$ , а також  $ED$  і  $EB$  задовольняють обмеження "NameAndTypeOK", то обмеження MCM ( \* мети і обмеження ідентичності MCM), що застосовується до  $ED$  повині бути:

1. такі ж, як ті, на які ЕВ в разі виведення поширюється, та
2. так само, або більш обмежені, ніж ті, на які ЕВ в разі виведення обмежується.

МСМ визначає таку поведінку для забезпечення того, щоб ніхто не зміг позбавитися від обмежень МСМ на елементах в комплексному типі на інший більш простий тип, що впливає з даного типу.

Забезпечення дотримання цієї умови з виведенням обмеження потребує виконання відповідних обмежень, з тим щоб оцінити стан двох вищеперерахованих. Такий рівень підтримки не передбачений більшістю основ XML Схем; Таким чином, більшість валідаторів МСМ, в іншому випадку необхідно дублювати великі частини компіляції логічних схем XML, щоб переконатися в послідовному використанні обмежень щодо виведення обмежень МСМ. Для того щоб зменшити цей тягар на реалізацію моделі валідаторів, МСМ вимагає, щоб всі елементи декларації з заданою назвою, які включені в комплекс типу визначення, повинні мати ті ж самі значення обмеження МСМ. Це дозволяє моделі валідаторів знайти обмежену частинку для обмеженої частинки, використовуючи простий збіг назв.

Це також означає, що вартість даного SML обмежень, застосованих до всіх заяв елемента із заданим ім'ям у комплексному визначенні типу можна розглядати як логічне доступні в одному місці, наприклад в нерухомість, який додається до цього типу комплексу, а не розсіяною за заявами в елементі даного типу. Наступний розділ використовує цю логічну зору, оскільки вона робить його більш зрозумілим і формально визначити SML обмеження поведінки через складний висновок типу.

## 5.4.2 Формальне визначення

### 5.4.2.1 Властивості

МСМ визначає чотири властивості для кожного комплексного типу визначення схеми СТ.

#### **{список обмежень необхідності мети}**

Список (Qімен, значень) пар, де

1. QName є кваліфікована назва ({namespace name} + {name}).
2. Значення властивостей **{необхідності мети}**.

#### **{список обмежень елемента мети}**

Список (Qімен, значень) пар, де

1. QName є кваліфікована назва ({namespace name} + {name}).
2. Значення властивостей **{елементу мети}**.

### {список обмежень типу мети}

Список (Qімен, значень) пар, де

1. QName є кваліфікована назва ({namespace name} + {name}).
2. Значення властивостей {типу мети}.

### {список обмежень ідентичності}

Список (Qімен, значень) пар, де

1. QName є кваліфікована назва ({namespace name} + {name}).
2. Значення властивостей {Визначення обмеження ідентичності MCM}

Значення вищеперерахованих 4 властивостей `xs:anyType` є порожнім.

#### 5.4.2.2 Обмеження конструкцій MCM

Присвоємо:

- CT = комплекс типу визначення.
- C = обмеження MCM (одне з {необхідності мети}, {елементу мети}, {типу мети}, { обмеження ідентичності MCM})
- P = властивість CT відповідна до обмеження C (одна з [{списку обмеження необхідності мети}](#), [{списку обмеження елементу мети}](#), [{спичку обмеження типу мети}](#), [{списку обмеження ідентичності }](#))
- V = значення P, список (QName, значень) пар.
- ED = елемент декларації, що міститься у CT.
- PED = властивість ED відповідних обмежень C (одна з [{необхідності мети}](#), [{елементу мети}](#), [{типу мети}](#), [{визначення обмеження ідентичності MCM}](#))

Властивість P - це призначене значення V, як вказано нижче:

1. Для кожного елемента декларації ED з кваліфікованою Qнавою, розміщених в CT:

А. Якщо ED не має обмежень C, тобто, значення PED відсутні (або помилкові у разі необхідності мети), то пропускаємо ED.

В. В іншому випадку, якщо вже є запис у V для Qназви, то пропускаємо ED.

**Примітка:**

Якщо значення існуючої запису відрізняється від значення PED, то попереднє розглядається як помилка схеми валідатору, як це зазначено в розділі [5.1.2.2 Схема компонентів правил](#) і в розділі [5.2.1.2 Правила компонентів схеми](#).

С. В іншому випадку, запис (Qназва, значення PED) додається до списку V.

2. Серед відповідних випадку застосовується наступне:

А. Якщо СТ визначається шляхом розширення простого типу визначенням, то значення V порожнє.

В. Якщо СТ визначається шляхом розширення комплексного типу визначення VT:

Початкове значення V обчислюється як це зазначено у списку пункту 1, вищеперерахованому, а потім

Для кожного запису (Qназва, VB) в значенні P в VT:

I. Якщо V має запис (Qназва, vC), то забезпечується VC, як таке ж як VB. Якщо це не так, то попереднє розглядається як помилка схеми валідатора.

II. Якщо V не має запису (Qназва, vC), то копіюємо (Qназва, VB) у V.

С. Якщо СТ визначається шляхом обмеження із комплексного типу визначення VT:

Початкове значення V обчислюється як це зазначено у списку пункту 1 вищеперерахованому, а потім

Для кожного запису (Qназва, VB) в значенні P в VT:

I. Якщо V має запис (Qназва, vC), то забезпечується VC, як допустиме обмеження VB, як це визначено в розділі [5.3 Чинні обмеження значень обмеження MCM](#). Якщо це не так, то попереднє розглядається як помилка схеми валідатора.

II. Якщо V не має запису (Qназва, VC), то копіюємо (Qназва, VB) у V.

### 5.4.2.3 Інстанції дії правил

Присвоємо:

- СТ = комплекс типу елемента декларації ED.
- E = приклад ED.
- C = дочірній елемент E.

Якщо C – елемент декларації, що містяться в СТ, і якщо один або декілька властивостей обмежень СТ, які зазначені в [5.4.2.1 Властивості](#), містить запис відповідності СТ з

кваліфікованою назвою ({namespace name} + {name}), то значення кожного з цих записів, що використовуються для оцінки відповідних обмежень на C, як визначено в розділі [5.1.2.3 інстанції дії Правил](#), як якщо б відповідний елемент декларації мав би відповідне обмеження з цим значенням.

### Примітка:

Один зі способів обмеження для вбудовування в елементі декларацій або визначення типів у схемі для обмеження елементів повинні бути включені в схему документа, вбудованих у відповідному місці в межах `xs:element` або `xs:complexType` елементах, які описують декларації елемента або тип визначення.

Елемент декларації та визначення типу, створених іншими засобами однак можуть також мати обмеження, закладені у (застосування інформації) їх властивостей (анотацій). Так, як це досягається, таке вкладення виходить за рамки цього стандарту і можуть варіюватися серед моделей процесорів.

## 6. Правила

XML Схема підтримує цілий ряд вбудованих у граматиці основних обмежень, але вона не підтримує мови для визначення довільних правил для стримування структури і змісту документів. Схематрон [\[ISO / IEC 19757-3\]](#) є ISO / IEC для тверджень про визначення набору документів XML. МСМ використовує Схематрон, щоб додати підтримку для додаткових обмежень моделі, які не підтримуються в XML схемі.

### 6.1 Неформальний опис (ненормативний)

У цьому розділі передбачається, що читач ознайомлений з концепцією Схематрона; стандартний Схематрон, що описан в [\[ISO / IEC 19757-3\]](#) і [\[Введення в Схематрон, поліпшення перевірки завдяки схематрону\]](#), є хорошим уроком по старій версії Схематрона.

Обмеження можуть бути вказані за допомогою `sch:assert` та `sch:report` елементів зі Схематрону. Наступний приклад використовує `sch:assert` елементи, щоб вказати два обмеження:

1. IPv4-адреса повинна мати чотири байти
2. IPv6-адреса повинна мати шістнадцять байт

```
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema" targetNamespace="urn:x-example:IPAddress">
  <xs:simpleType name="IPAddressVersionType">
    <xs:restriction base="xs:string" >
      <xs:enumeration value="V4" />
      <xs:enumeration value="V6" />
    </xs:restriction>
  </xs:simpleType>

  <xs:complexType name="IPAddress">
    <xs:annotation>
      <xs:appinfo>
```

```

    <sch:schema xmlns:sch="http://purl.oclc.org/dsdl/schematron">
      <sch:ns prefix="tns" uri="urn:x-example:IPAddress" />
      <sch:pattern id="Length">
        <sch:rule context=".">
          <sch:assert test="tns:version != 'V4' or
count(tns:address) = 4">
            A v4 IP address must have 4 bytes.
          </sch:assert>
          <sch:assert test="tns:version != 'V6' or
count(tns:address) = 16">
            A v6 IP address must have 16 bytes.
          </sch:assert>
        </sch:rule>
      </sch:pattern>
    </sch:schema>
  </xs:appinfo>
</xs:annotation>
<xs:sequence>
  <xs:element name="version" type="tns:IPAddressVersionType" />
  <xs:element name="address" type="xs:byte" minOccurs="4"
maxOccurs="16" />
</xs:sequence>
</xs:complexType>
</xs:schema>

```

**Обмеження схематрона**, вбудований в `xs:annotation/xs:appinfo` елемент для визначення комплексних типів визначення або елементів декларації, поширюється на всі приклади комплексного типу або елемента. У наведеному вище прикладі, модель довжини (що є частиною обмежень Обмеження Схематрона) застосовується у всіх елементах, тип яких є IP-адреса, або похідний тип IP-адреси. Шаблон елемента містить одну або більше `sch:rule` елемент і один `sch:rule` елемент, що містить одну чи більше `assert` і/або `report` елементи. Кожен `sch:rule` елемент визначає свої зв'язки з використанням контекстного атрибуту. Цей контекстний вираз обчислюється в контексті кожного елемента і застосовних результатів у вузлі елемента множини, для якого стверджується протокол випробувань і виразів, що містяться в `sch:rule` елементі оцінки. Контекст виразу визначається як шаблон XSLT. Це означає, що `smlfn:deref` функція не може бути використана контексті слова.

У наведеному вище прикладі, `context="."`. Тому стверджуємо, що два вирази оцінюються в контексті кожного діючого елемента, тобто кожен елемент типу IP-адреси. Вираз для випробувань стверджується як логічне вираження, а також стверджується, порушується (або сгоряє), якщо його вираз випробування не справджується. Доповідь порушена (або спалена), якщо його вираз випробування справджується. Таким чином, ствердження можуть бути перетворені в доповіді на просте заперечення свого виразу випробування. Наступний приклад використовує доповідь елемента, що являють собою IP адресу обмежень попереднього прикладу:

```

<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema" targetNamespace="urn:x-
example:IPAddress">
  <xs:simpleType name="IPAddressVersionType">
    <xs:restriction base="xs:string">
      <xs:enumeration value="V4"/>
      <xs:enumeration value="V6"/>
    </xs:restriction>
  </xs:simpleType>

```



```

<xs:complexType name="IPAddress">
  <xs:annotation>
    <xs:appinfo>
      <sch:schema xmlns:sch="http://purl.oclc.org/dsdl/schematron">
        <sch:ns prefix="tns" uri="urn:x-example:IPAddress" />
        <sch:pattern id="Length">
          <sch:rule context=".">
            <sch:report test="tns:version = 'V4' and
count(tns:address) != 4">
              A v4 IP address must have 4 bytes.
            </sch:report>
            <sch:report test="tns:version = 'V6' and
count(tns:address) != 16">
              A v6 IP address must have 16 bytes.
            </sch:report>
          </sch:rule>
        </sch:pattern>
      </sch:schema>
    </xs:appinfo>
  </xs:annotation>
  <xs:sequence>
    <xs:element name="version" type="tns:IPAddressVersionType" />
    <xs:element name="address" type="xs:byte" minOccurs="4"
maxOccurs="16" />
  </xs:sequence>
</xs:complexType>
</xs:schema>

```

Якщо `sch:assert` або `sch:report` порушуються, порушення повідомляється разом з зазначеним повідомленням. Повідомлення може включати заміщення рядків на основі XPath виразів. Вони можуть бути визначені за допомогою `sch:value-of` елемента. Наступний приклад використовує `sch:value-of` елемент, щоб включити вказану кількість байт адреси в повідомлення:

```

<sch:assert test="tns:version != 'v4' or count(tns:address) = 4">
  A v4 IP address must have 4 bytes instead of the specified
  <sch:value-of select="string(count(tns:address))"/> bytes.
</sch:assert>

```

На додаток, щоб бути вбудованими в комплексний тип визначення, обмеження можуть бути вбудовані в глобальний елемент декларації. Такі обмеження оцінюються для кожного прикладу відповідного елемента до глобального елемента декларації. Наведемо приклад:

```

<xs:element name="StrictUniversity" type="tns:UniversityType">
  <xs:annotation>
    <xs:appinfo>
      <sch:schema xmlns:sch="http://purl.oclc.org/dsdl/schematron">
        <sch:ns prefix="u" uri="http://www.university.example.org/ns" />
        <sch:ns prefix="smlfn"
          uri="http://www.w3.org/ns/sml-function"/>
        <sch:pattern id="StudentPattern">
          <sch:rule context="u:Students/u:Student">
            <sch:assert test="smlfn:deref(.)[starts-
with(u:ID, '99')]">
              The specified ID <sch:value-of
select="string(u:ID)"/>
              does not begin with 99.
            </sch:assert>
          </sch:rule>
        </sch:pattern>
      </sch:schema>
    </xs:appinfo>
  </xs:annotation>
</xs:element>

```

```

        </sch:assert>
        <sch:assert test="count(u:Courses/u:Course)>0">
            The student <sch:value-of select="string(u:ID)"/>
must be enrolled
            in at least one course.
        </sch:assert>
    </sch:rule>
</sch:pattern>
</sch:schema>
</xs:appinfo>
</xs:annotation>
</xs:element>

```

`sch:rule` елементи, що містяться в `StudentPattern` застосовні до всіх прикладів елементів глобального `StrictUniversity` елемента декларації. Для кожного елемента `StrictUniversity`, вираз XPath вказаний як значення атрибуту контексту оціненого, щоб повернути набір вузлів, а вирази випробування для двох стверджувальних оцінюються для кожного вузла в цьому наборі вузлів. Таким чином, ці два ствердження перевіряють наступні умови для кожного прикладу `StrictUniversity`.

1. ID кожного учня повинно починатися з '99'.
2. Кожному студенту має бути зарахований принаймні один курс.

Шаблони схематрона можуть бути авторами в окремих правилах документів, які потім обов'язково комплектують документ в модель.

Нижче наведено приклад обмежень для `StrictUniversity`, виражених в окремому документі:

```

<?xml version="1.0" encoding="utf-8" ?>
<sch:schema xmlns:sch="http://purl.oclc.org/dsdl/schematron">
  <sch:ns prefix="u" uri="http://www.university.example.org/ns" />
  <sch:ns prefix="smlfn" uri="http://www.w3.org/ns/sml-function" />
  <sch:pattern id="StudentPattern">
    <sch:rule context="u:StrictUniversity/u:Students/u:Student">
      <sch:assert test="smlfn:deref(.)[starts-with(u:ID,'99')]">
        The specified ID <sch:value-of select="string(u:ID)"/>
        does not begin with 99.
      </sch:assert>
      <sch:assert test="count(u:Courses/u:Course)>0">
        The student <sch:value-of select="string(u:ID)"/> must be
enrolled
        in at least one course.
      </sch:assert>
    </sch:rule>
  </sch:pattern>
</sch:schema>

```

Обов'язковим для правила документа, що містить `StudentPattern` шаблон до документів, які можуть містити екземпляри елемента `StrictUniversity`, визначається реалізацією.

## 6.2 Правила підтримки

Моделі валідаторів необхідні для підтримки та оцінки XPath виразів, які поєднуються з

`smlfn:deref()` функцією в тілі схематрона обмежень.

Якщо атрибут `queryBinding` не вказаний, то його значенню належить бути встановленим на "XSLT". Моделі валідаторів повинні обов'язково підтримувати "XSLT" запит. Моделі валідаторів можуть додатково підтримувати запити прив'язки окрім "XSLT".

## 6.3 Правила, пов'язані з компонентами схеми

MCM визначає нові властивості для кожного компонента схеми комплексного типу визначення і кожного елемента компонента схеми декларації.

### (Правила)

Набір обмежень схематрон.

#### 6.3.1 Правила конструкцій MCM

{Правила} властивості містить всі обмеження схематрона і застосовуються щодо прикладів данного типу визначення або елемента декларації. Його значення визначається в частині `sch:schema` елементів в рамках компоненту, а іноді і в частині з властивостей {Правил} інших компонентів.

`sch:schema` елементи можуть з'являтися як елементи в {застосуванні інформації} з {анотації} глобального елемента декларації або глобального типу визначення. Цій специфікація надається жодне значення `sch:schema` елементу, якщо воно з'являються в якості пунктів у будь-якому іншому місці.

Нехай **місцеві правила** з урахуванням глобального елемента декларації або глобального комплексного типу визначення є безліч обмежень Схематрона, вбудованих в програму інформації властивості схематичного елемента (анотації). Для інших компонентів **місцеві правила** є порожніми.

Значення властивостей {Правил} для компонента схеми розраховується наступним чином:

1. Значення {Правила} для XS: AnyType є порожньою множиною.
2. Якщо схема компоненту це глобальний елемент декларації, то значення її {Правила} є об'єднанням її **місцевих правил** і відповідним прикладом з наступних функцій:
  - A. Якщо елемент декларації має {приналежність до групи}, то значення {Правила} з {приналежність до групи заміщення}.
  - B. В іншому випадку (елемент декларації не має {приналежність до групи}), множина порожня.
3. Якщо схема компонентів є комплексним типом визначення, то значення властивостей її {Правил} є об'єднанням її **місцевих правил** і відповідним

прикладом з наступних функцій:

А. Якщо компонент {базового типу визначення} є комплексом типу визначення, то {Правила} є {базового типу визначення}. Це справджується для виведення в більш широкому сенсі, а також для виведення обмеженням.

В. В іншому випадку (коли {базовий тип визначення} є простим типом визначення), множина порожня.

4. В іншому випадку, значення) {Правила} власності не впливає на цю специфікацію.

### 6.3.2 Схема компонентів правил

Моделі валідаторів повинні втілювати наступні правила.

1. Якщо комплексний тип **D** відбувається звуженням або розширенням бази від {базового типу визначення} **B**, і якщо **B** містить певні {Правила} на ньому, то вони повинні автоматично копіюватися в **D** і об'єднуватися в певні {Правила} на **D**.

2. Якщо комплексний тип **D** відбувається звуженням або розширенням бази від {базового типу визначення} **B**, то глобальний елемент декларації з непорожніми {Правилами}, що містяться в **B**, повинні не обмежуватися місцевими елементами декларації в **D**.

#### Примітка:

Спостерігається помилка, якщо всі наступні умови справджуються.

А. Елемент декларації **ED** міститься (прямо, опосередковано або неявно) в **D** і елемент декларації **EB** міститься (прямо, опосередковано або неявно) в **B**.

В. **ED** та **EB** задовольняють "обмеження NameAndTypeOK" (для XML схеми визначення дійсного обмеження, див. [схему компонентів обмеження: частинне обмеження, обмеження на частку компонентів схеми в \[XML Структурі схеми\]](#)).

С. **EB** - це посилання на глобальний елемент декларації з обмеженням схематрону на ньому.

Д. **ED** є місцевим елементом декларації з тією ж назвою, що і **EB**.

### 6.3.3 Приклад правил переконливості

Моделі валідаторів повинні вести себе таким чином:

1. Кожне [обмеження схематрону](#) в {Правилах} комплексного типу визначення

**СТ** повинне бути оцінене для всіх прикладів елемента елемента типу **СТ** в моделі під час перевірки валідатора.

2. Кожне **обмеження схематрона** в **{Правилах}** глобального елемента декларації **G** повинне бути оцінене для всіх прикладів елемента **G** в моделі під час перевірки валідатора.

3. Усі перевірені твердження випускають поле правил, які повинні увінчатися успіхом.

## 6.4 Авторські правила в документах правил

### 6.4.1 Обов'язкове правило

**Моделі валідаторів** повинні забезпечити механізм для підтримки обов'язковості моделей Схематрона, авторизуються у вигляді окремих **документах правил** для набору документів в моделі. Документи правил можуть бути пов'язані з моделлю прикладів документів, відомі як документи моделі визначення. Механізм документів обов'язкових правил моделі визначається реалізацією.

## 7. Локалізація природної мови повідомлень

МСм визначає `sml:locid` атрибути на підтримку локалізації текстів природної мови, наприклад, `smlif:description` або повідомлення схематрону. **Моделі валідаторів** можуть підтримувати `sml:locid` атрибут за наступним елементам:

1. `sch:assert` та `sch:report` документи правил.
2. `sch:assert` and `sch:report` у шаблоні схематрону вбудовані в програмну інформацію анотації властивостей комплексного типу визначення або елемента декларації.
3. Елементи в прикладах документів з текстовим вмістом.

Моделі валідаторів, що підтримуються `sml:locid` атрибутами, повинні використовувати значення `sml:locid` атрибутів для доступу місцезрештування перекладеного тексту.

### Примітка:

Механізм використання QName значення `sml:locid` атрибуту забезпечує знаходження перекладеного тексту в залежності від реалізації. Наприклад, простір імен може бути використаний для ідентифікації ресурсу, що містить текст і місцева назва може бути використана для ідентифікації тексту в таких ресурсах. Див **Додаток F. локалізації та підстановка змінних Зразків** для конкретних прикладів того, як SML: locid атрибут може бути використаний для підтримки текстових локалізації.

### 7,1 Підстановка змінних

Часто буває, що `sch:assert` or `sch:report` повідомлення можуть бути використані у

різних ситуаціях. Щоб мати можливість повторного використання повідомлень, верховенство автора має бути в змозі замінити змінним змістом, заснованим на контексті, в якому повідомлення буде використовуватися.

Хоча ця специфікація не зобов'язує використовувати підстановку змінних в повідомленнях схематрону, він припускає використання `xsl:variable`, коли підстановка змінних є бажаною. Див [Додаток F. локалізації та підстановка змінних Зразків](#), розділ для конкретного прикладу того, як `xsl:variable` може бути використана на підтримку повторного використання локалізованих повідомлень.

## 8. Відповідність критеріям

Програма є відповідною для моделі процесора MCM, якщо і тільки якщо всі обмеження задовольняють процесори в інших частинах цієї специфікації.

Відповідної модель MCM процесора є відповідною моделлю валідатора MCM, якщо і тільки якщо вона задовольняє такі умови:

1. Валідатор повинен виконувати перевірки моделі, як це визначено в цій специфікації.
2. Валідатор повинен підтримувати XML 1.0 [\[XML\]](#), XML схеми 1.0 [\[Структури XML схем, XML схеми типів даних\]](#) і XPath 1.0 [\[XPath\]](#), але може також додатково підтримуватися будь-якими майбутніми версіями цих специфікацій.
3. Валідатор повинен підтримувати Схематрон [\[ISO / IEC 19757-3\]](#).
4. Валідатор повинен виконувати правила оцінки схематрону на етапі # BCE.
5. Валідатор повинен підтримувати `deref ()` XPath функцію розширення.
6. Валідатор повинен ідентифікувати всі посилання MCM в моделі, використовуючи посаду перевірка схеми Інфосет. [\[XML схематична Структура\]](#)
7. Валідатор повинен використовувати посадові перевірки схем Інфосет, щоб визначити посилання MCM в моделі невизначеного посилання MCM. [\[XML Схематична Структура\]](#)

Відповідність моделі і дійсності моделі може бути оцінена, якщо і тільки якщо всі документи в моделі доступні моделі валідатору. Модель валідатора зобов'язана документувати її поведінку, коли буде знайдена модель документа, який буде недоступним (тобто поведінка визначається реалізацією). Вона може реагувати на цю умову тими способами, які включають, але не обмежуються ними: оцінки моделі недійсні або ліквідуються, про що попереджалось. Мета широти надається моделі валідатора в даному випадку в наданні деяких гнучких здійснень, які не пропонують обмежений набір варіантів, однак який буде тлумачитися вузько, а не широкою ліцензією, щоб прийняти заходи, пов'язані з незабезпеченням обмежень MCM на пов'язанні документів.

Модель являє собою відповідну модель MCM, якщо і тільки якщо вона задовольняє

Наступні умови:

1. Кожен [документ](#) в моделі повинен бути добре сформованими XML-документом [\[XML\]](#)
2. Для кожного документа XML схема у визначенні документу моделі [дієва] властивості кореневого елемента має бути "дійсною", коли схема дії оцінюється по відношенню до схеми, побудованої з [XML-схеми для схем і А. Нормативних схем](#) документів схеми.
3. Усі схеми зібрані з документів XML-схем у визначенні документу моделі повинні задовольняти умовам, вираженим в Помилках конструкцій схем і структур (§ 5.1). [\[XML Схематичні Структури\]](#)
4. Усі схеми зібрані з документів XML-схем у визначенні документу моделі повинні задовольняти умовам, висловленим в розділах [5.1.1.1 Обмеження конструкцій МСМ](#), [5.1.1.2 Правила схематичних конструкцій МСМ](#) [5.1.2.1 Обмеження конструкцій МСМ](#) [5.1.2.2 Правила схематичних конструкцій МСМ](#) [5.2. 1,1 Обмеження конструкцій МСМ](#) [5.2.1.2 Правила схематичних конструкцій МСМ](#) [6.3.1 Обмеження конструкцій МСМ](#) [6.3.2 Правила схематичних конструкцій МСМ](#).
5. Кожен документ схематрну у визначенні документу моделі повинен бути дійсними документом схематрона [\[ISO / IEC 19757-3\]](#)

**Примітка:**

Ця специфікація не визначає, якими схемами зібрані і які документи схеми сприяють складанню схеми.

Відповідної модель МСМ є дійсною, якщо і тільки якщо вона відповідає всім наступним умовам:

1. У кожному прикладі документа в моделі, яка прив'язана до схеми [дієві] властивості кореневого елемента мають бути "дійсними", і [дієві] надбання всіх інших елементів і всіх атрибуту не повинні бути "недійсним", коли схема дії оцінюється по відношенню до будь-якої схеми, прив'язаною до цього прикладу документу. Оцінка схематичних дій починається не з передбаченої Декларації або визначення в кореновому елементі. [\[XML Схематична Структури\]](#)

Схеми дій прикладів документів не прив'язані до якоїсь схеми і не сприяють здійсненню або нездійсненню моделі.

**Примітка:**

Як схема межує з прикладом документів є невизначеним в даній специфікації. Кілька схем можуть бути прив'язані до цього прикладу документу.

Дія МСМ тягне за собою не існування працездатності схеми в кореновому або будь-якому похідному. Дія МСМ насправді може бути не дарма перевіреною

тільки після оцінки дії схем, і тільки на похідних ділянках, для яких PSVI доступний.

Оскільки глибина PSVI залежить від реалізації, є відмінність в обмеженні видимості обмежень MCM, доступна для валідаторів MCM, завдяки чому отримується результат дії MCM

2. Кожен документ в модель повинен задовольнити всі чинні обмеження схематрона під час перевірки фази # BCE.

3. Кожен документ в моделі повинен задовольнити всі нормативні акти в цій специфікації, які відносяться до моделі документів.

#### Примітка:

Це означає, наприклад, що кожен документ повинен відповідати всім застосовним ациклічним MCM, цільовим MCM, а також обмеженням ідентичності MCM.

## 9. Посилання Розширення MCM (ненормативні)

Даний розділ являє собою довідник для розширень MCM XML-схеми і XPath.

### 9.1 Атрибути

#### 9.1.1 sml:acyclic - ациклічні

Використовується для вказівки прикладів посилань MCM даного типу і його похідних типів, не створюючи ніяких циклів в моделі

```
<xs:attribute name="acyclic" type="xs:boolean"/>
```

Якщо цей атрибут встановлено вірно для комплексного типу **СТ**, то приклади **СТ** (включаючи будь-які похідні типи **СТ**), який є посиланнями MCM, не може створювати будь-які цикли в моделі. У наступному прикладі, HostedOnRefType є комплексним типом декларації, екземпляри якого не можуть створювати цикл:

```
<xs:complexType name="HostedOnRefType" sml:acyclic="true">  
...  
</xs:complexType>
```

Якщо sml:acyclic атрибут не зазначено або визначено помилковим для комплексного типу декларацій, то приклади подібного типу, які являються посиланнями MCM, можуть створювати цикли в моделі.

#### 9.1.2 sml:ref - посилальні

Цей глобальний атрибут використовується для визначення посилання MCM.

```
<xs:attribute name="nilref" type="xs:boolean"/>
```



Будь-який елемент, який має `sml:ref="true"`, буде розглядатися як посилання MCM.

### 9.1.3 `sml:nilref` – нульові посилання

Цей глобальний атрибут використовується для визначення нульове посилання MCM.

```
<xs:attribute name="nilref" type="xs:boolean"/>
```

Будь-яке посилання MCM, яке має `sml:nilref="true"` or `sml:nilref="1"` буде трактуватися як нульове посилання MCM.

### 9.1.4 `sml:targetElement` – елементи мети

QName представляє назву посилального елемента

```
<xs:attribute name="targetElement" type="xs:QName"/>
```

`sml:target` елемент підтримується як атрибут для будь-якого елемента декларації. Значення цього атрибута повинно бути назвою якогось глобального елемента декларації. Нехай `sml:targetElement="ns:GTE"` для деякого елемента декларації **E**. Тоді кожен елемент інстанції **E** повинен бути орієнтований на елемент, який є прикладом **ns:GTE** або прикладом деякого глобального елемента декларації, в ієрархічній групі якого головним є **ns:GTE**.

У наступному прикладі елемент, який посилається на приклад `HostOS`, повинен бути прикладом `win:Windows`:

```
<xs:element name="HostOS" type="tns:HostOSRefType"
sml:targetElement="win:Windows" minOccurs="0"/>

<xs:complexType name="HostOSRefType">
  <xs:sequence>
    <xs:any namespace="##any" processContents="lax" minOccurs="0"
maxOccurs="unbounded"/>
  </xs:sequence>
  <xs:anyAttribute namespace="##any" processContents="lax"/>
</xs:complexType>
```

Модель є недійсною, якщо її документи порушують одне чи більше SML: `targetElement` обмеження.

### 9.1.5 `sml:targetRequired`

Використовується для вказівки прикладів, де посилання MCM повинні бути орієнтованими на елементи в моделі, тобто, наприклад посилання MCM не може бути порожнім або містити недозволені посилання. Тому вважається за помилку, якщо вказано `targetRequired="true"` на елементі декларації, в якій відповідний елемент посилання MCM **R** має `sml:nilref="true"`

```
<xs:attribute name="targetRequired" type="xs:boolean"/>
```

У наступному прикладі `targetRequired` атрибут використовується для вказівки, де приклад повинен бути застосованим операційною системаю.

```
<xs:complexType name="ApplicationType">
  <xs:sequence>
    <xs:element name="Name" type="xs:string"/>
    <xs:element name="Vendor" type="xs:string"/>
    <xs:element name="Version" type="xs:string"/>
    <xs:element name="HostOSRef" type="tns:HostOSRefType"
sml:targetRequired="true"/>
  </xs:sequence>
</xs:complexType>

<xs:complexType name="HostOSRefType">
  <xs:sequence>
    <xs:any namespace="##any" processContents="lax" minOccurs="0"
maxOccurs="unbounded"/>
  </xs:sequence>
  <xs:anyAttribute namespace="##any" processContents="lax"/>
</xs:complexType>
```

Модель є недійсною, якщо її документи порушують одне чи більше SML: `targetRequired` обмеження.

### 9.1.6 `sml:targetType` – тип мети

`QName` представляє тип посилального елемента

```
<xs:attribute name="targetType" type="xs:QName" />
```

`sml:targetType` підтримується як атрибут для будь-якого елемента декларацій. Якщо значення цього атрибута визначається як **T**, потім тип посилального елемента повинен бути або **T**, або похідним від типу **T**. В наступному прикладі тип елемента, на який посилається елемент Операційної системи повинен бути `ibm:LinuxType`, або його похідного типу

```
<xs:element name="OperatingSystem" type="tns:OperatingSystemRefType"
sml:targetType="ibm:LinuxType" minOccurs="0"/>

<xs:complexType name="OperatingSystemRefType">
  <xs:sequence>
    <xs:any namespace="##any" processContents="lax" minOccurs="0"
maxOccurs="unbounded"/>
  </xs:sequence>
  <xs:anyAttribute namespace="##any" processContents="lax"/>
</xs:complexType>
```

Модель є недійсною, якщо її документи порушують одне чи більше SML: `TargetType` обмеження.

### 9.1.7 `sml:locid`

Цей атрибут може бути визначений на `sch:assert`, `sch:report` і на будь-який елемент текстового змісту. SML: `locid` атрибут використовується для визначення

місцерозташування перекладу в змісті тексту, що містить даний елемент.

```
<xs:attribute name="locid" type="xs:QName" />
```

Механізм використання значення QName `sml:locid` атрибута для пошуку перекладеного тексту здійснюється специфічно, а, отже, виходить за рамки цієї специфікації.

## 9.2 Елементи

### 9.2.1 Ключові елементи MCM, `sml:key`

Цей елемент використовується для вирішення основних перешкод у деяких сферах. Семантика – це те ж саме, що і `xs:key` за винятком того, що `sml:key` також може використовуватися для визначення основних перешкод в інших документах, тобто `sml:selector` - дочірній елемент `sml:key` може містити `deref` функції для вирішення елементів, викладених в іншому документі .

```
<xs:element name="key" type="sml:keybase" />
```

`sml:key` підтримується в `appinfo` Від `xs:element`.

### 9.2.2 Ключові посилання MCM, `sml:keyref`

Застосовує обмеження, що містить в контексті `xs:element`, в спектрі діапазону якого викладений посиланням документ.

```
<xs:element name="keyref">
  <xs:complexType>
    <xs:complexContent>
      <xs:extension base="sml:keybase">
        <xs:attribute name="refer" type="xs:QName" use="optional"/>
      </xs:extension>
    </xs:complexContent>
  </xs:complexType>
</xs:element>
```

### 9.2.3 Унікальні елементи MCM, `sml:unique`

Цей елемент використовується для визначення унікальності обмеження в деяких сферах. Семантика – це те ж саме, що і `sml:unique` за винятком того, що `sml:unique` також може бути використаний для вказівки обмежень на унікальність інших документів, тобто `sml:selector` - дочірній елемент `sml:unique` може містити `deref` функції для вирішення елементів в іншому документі .

```
<xs:element name="unique" type="sml:keybase" />
```

`sml:unique` підтримується в `appinfo` Від `xs:element`.

### 9.2.4 URI елементи MCM, `sml:uri`

Вказує посилання MCM, наприклад URI посилання Схеми MCM.

```
<xs:element name="uri" type="xs:anyURI" />
```

Цей елемент повинен бути використаний для вказівки посилання MCM, яка використовується URI посиланням схеми MCM.

## 9.3 XPath функції

### 9.3.1 smlfn: deref

```
node-set deref (node-set)
```

Ця функція приймає набір вузлів і намагається вирішити посилання MCM. Отриманий набір вузлів налічує безліч елементів, які отримані від успішного рішення (або де - посилання) посилання MCM. Наприклад,

```
deref (/u:Universities/u:Students/u:Student)
```

буде вирішувати посилання MCM - `Student`. Мета посилання MCM завжди повинна бути елементом.

## 10. Список літератури

### 10,1 Нормативна

#### [SML-IF 1.1]

[Мова Службового Моделювання, Обмінного Формату, Версія 1.1](#), редактори: Балчандра Пандіт, Валентин Попеску, Вірджинія Сміт. Світовий Широкий Мережний Консорціум від 12 травня 2009 року. З версією Мови Службового Моделюванню Обмінного Формату можна ознайомитися на <http://www.w3.org/TR/2009/REC-sml-if-20090512/>. Остання версія Мови Службового Моделюванню Обмінного Формату версії 1.1 доступна на <http://www.w3.org/TR/sml-if/>.

#### [IETF RFC 2119]

[Ключові слова для використання в RFC для позначення рівнів вимог](#), автор - К. Браннер. Інтернет споруджений для особливого призначення в червні 1999 року. Доступний за адресою: <http://www.ietf.org/rfc/rfc2119.txt>.

#### [IETF RFC 3986]

[Загальноприйнятий Ідентифікатор Ресурсу \(ZIP\): Загальний Синтаксис](#), автори: Т. Бернерс-Лі, Р. Філдінг, Л. Масінтер. Інтернет споруджений для особливого призначення в січні 2005 року. Доступний за адресою: <http://www.ietf.org/rfc/rfc3986.txt>.

#### [IETF RFC 3987]

[Інтернаціоналізовані ідентифікатори ресурсів \(IIR\)](#), автори: М. Дуерст, М. Сігард. Інтернет споруджений для особливого призначення в січні 2005 року. Доступний за адресою: <http://www.ietf.org/rfc/rfc3987.txt>.

#### [RFC 2234]

[Збільшений BNF для Специфікацій Синтаксису: ABNF](#), інтернет консорціум, листопад 1997 року. Наявність на <http://www.rfc-editor.org/rfc/rfc2234.txt>.

**[ISO / IEC 19757-3]**

[Інформаційні технології – Мова Визначення Документальних Схем \(МВДС\) – Частина 3: Правила базування на твердженні - Схематрон.](#) Міжнародна організація по стандартизації та Міжнародної електротехнічна комісія, 1 січня 2006 року. Наявність на [http://standards.iso.org/itf/PubliclyAvailableStandards/c040833\\_ISO\\_IEC\\_19757-3\\_2006](http://standards.iso.org/itf/PubliclyAvailableStandards/c040833_ISO_IEC_19757-3_2006) (E). Zip

**[XML- NS]**

[Простори імен в МСМ 1.0 \(друге видання\)](#), редактори: Тім Брей, Дейв Холландер, Ендрю Лойман, Річард Тобін. Світовий Широкий Мережний Консорціум від 16 серпня 2006 року. Дана версія Простору імен в МСМ доступна на <http://www.w3.org/TR/2006/REC-xml-names-20060816/>. Останню версію Простору імен в МСМ можна знайти на <http://www.w3.org/TR/xml-names/>.

**[XML]**

[Розширена Мова Націнки \(XML\) 1.0 \(четверте видання\)](#), редактори Т. Брей, Д. Паолі, СМ Сперберг-Маккуїн, Е. Малер. Світовий Широкий Мережний Консорціум від 10 лютого 1998 року, переглянута 16 серпня 2006. Видання, цитоване на (<http://www.w3.org/TR/2006/REC-xml-20060816>), було одним, що поширюється на дату публікації цієї специфікації як Кандидатської Рекомендації. Останній версія XML 1.0 доступна на <http://www.w3.org/TR/xml/>. Реалізації можуть слідувати виданню, цитованому і / або будь-якого більш пізнього видання, вони визначається реалізацією того видання, яке підтримує реалізацію.

**[XML Information Set]**

[Набір Інформацій XML \(друге видання\)](#), редактори: Джон Кован, Річард Тобін. Світовий Широкий Мережний Консорціум від 4 лютого 2004 року. Дана версія Набору Інформацій XML рекомендована на <http://www.w3.org/TR/2004/REC-xml-infoset-20040204/>. Остання версія Набору Інформацій XML доступна на <http://www.w3.org/TR/xml-infoset/>.

**[XML Schema Структури]**

[XML Схематична Частина 1: Структури \(друге видання\)](#), редактори: Н. Томпсон, Д. Бук, М. Малоні, М. Мендельсон. Світовий Широкий Мережний Консорціум від 2 травня 2001 року, переглянутий 28 жовтня 2004. Дана версія XML Сематичної Частини 1 рекомендована на <http://www.w3.org/TR/2004/REC-xmlschema-1-20041028>. Остання версія 1.0 XML Схематичної Частини 1 доступна на <http://www.w3.org/TR/xmlschema-1>.

**[XML Schema типи даних]**

[XML Сематична Частина 2: Типи Даних \(друге видання\)](#), редактори: Х. Байрон і А. Малхотра. Світовий Широкий Мережний Консорціум від 2 травня 2001 року, переглянутий 28 жовтня 2004. Дана версія XML Схематичної Частини 2 рекомендована на <http://www.w3.org/TR/2004/REC-xmlschema-2-20041028>. Останню версію 1.0 XML Схемтичної Частини 2 можна отримати на <http://www.w3.org/TR/xmlschema-2>.

**[XML-схеми для Схеми]**

[XML-схеми для XML-схем.](#) Світовий Широкий Мережний Консорціум від 2 травня 2001 року, переглянутий 28 жовтня 2004.

**[XPath]**

[Мова Шляху XML \(XPath\) Версія 1.0](#), редактори: Д. Кларк С. Дероуз. Світовий Широкий Мережний Консорціум від 16 листопада 1999 року. Дана версія XML Мови Шляху (XPath) Версія 1.0 доступна на <http://www.w3.org/TR/1999/REC-xpath-19991116>. Останню версію XML Мови Шляху (XPath) Версія 1.0 можна

переглянути на <http://www.w3.org/TR/xpath>.

#### [XPointer]

[Рамкова покажчик](#), редактори: П. Грос, Є. Малер, Д. Марш, Н. Уолш. Світовий Широкий Мережний Консорціум від 25 березня 2003 року. Дана версія Рамкового Покажчика рекомендований на <http://www.w3.org/TR/2003/REC-xptr-framework-20030325/> остання версія Рамкового Покажчика наявна на <http://www.w3.org/TR/xptr-framework/>.

#### [XMLNS () Схема]

[Схематичний Покажчик xmlns\(\)](#), редактори: С. Дероуз, р. Даніель молодший, Е. Малер, Дж. Марш. Світовий Широкий Мережний Консорціум від 25 березня 2003 року. Дана версія Схематичного Покажчика xmlns() рекомендована на <http://www.w3.org/TR/2003/REC-xptr-framework-20030325/>, останню версію Схематичний Покажчик xmlns() можна знайти на <http://www.w3.org/TR/xptr-xmlns/>.

## 10,2 Ненормативна

#### [Введення в Schematron]

[Введення в Схематорн](#), автор - Едді Робертсон. Орелей Медіа від 12 листопада 2003 року. Наявний на <http://www.xml.com/pub/a/2003/11/12/schematron.html>

#### [Вдосконалення схвалення Схематрона]

[Поліпшення перевірки документів XML за допомогою Схематрону](#), автор - Деір Обасанджо. Корпорація Microsoft, вересень 2004 року. Наявний на <http://msdn.microsoft.com/en-us/library/aa468554.aspx>

#### [XML Schema Primer]

[XML Сематична Частина 0: Підручник для початківців](#), Друге видання, редактори Д. Фолсайд, П. Уолмслі. Світовий Широкий Мережний Консорціум від 2 травня 2001 року, переглянутий 28 жовтня 2004. Дана версія XML Схематичної Частини 0 рекомендована на <http://www.w3.org/TR/2004/REC-xmlschema-0-20041028>. Остання версія XML Схематичної Частини 0 доступна на <http://www.w3.org/TR/xmlschema-0>.

## A. Нормативна схема MSM

```
<!--
/*
 * Copyright © ns World Wide Web Consortium,
 *
 * (Massachusetts Institute of Technology, European Research Consortium for
 * Informatics and Mathematics, Keio University). All Rights Reserved. This
 * work is distributed under the W3C® Document License [1] in the hope that
 * it will be useful, but WITHOUT ANY WARRANTY; without even the implied
 * warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.
 *
 * [1] http://www.w3.org/Consortium/Legal/2002/copyright-documents-20021231
 */
--><xs:schema
  xmlns:sml="http://www.w3.org/ns/sml"
  xmlns:xs="http://www.w3.org/2001/XMLSchema"
  targetNamespace="http://www.w3.org/ns/sml"
  elementFormDefault="qualified"
  blockDefault="#all"
```

```

version="1.0"
xml:lang="en"
finalDefault=""
attributeFormDefault="unqualified">
<!--
  References
  =====
-->

<!-- CONTEXT: To be used in any <xs:element> -->
<xs:attribute name="ref" type="xs:boolean">
  <xs:annotation>
    <xs:documentation>
      Specifies if the element contains a reference
    </xs:documentation>
  </xs:annotation>
</xs:attribute>
<!-- CONTEXT: To be used in any <xs:element> -->
<xs:attribute name="nilref" type="xs:boolean">
  <xs:annotation>
    <xs:documentation>
      Specifies that the reference element denotes a "null" reference.
      To be used only on elements for which sml:ref="true".
    </xs:documentation>
  </xs:annotation>
</xs:attribute>
<!-- CONTEXT: To be used in any <xs:element> -->
<xs:attribute name="targetElement" type="xs:QName">
  <xs:annotation>
    <xs:documentation>
      A qualified name of a global element in the referenced document.
    </xs:documentation>
  </xs:annotation>
</xs:attribute>
<!-- CONTEXT: To be used in any <xs:element>-->
<xs:attribute name="targetRequired" type="xs:boolean">
  <xs:annotation>
    <xs:documentation>
      If true, requires the target element of the reference to
      exist in the model.
    </xs:documentation>
  </xs:annotation>
</xs:attribute>
<!-- CONTEXT: To be used in any <xs:element>-->
<xs:attribute name="targetType" type="xs:QName">
  <xs:annotation>
    <xs:documentation>
      A qualified name of the type of the element in the
      referenced document.
    </xs:documentation>
  </xs:annotation>
</xs:attribute>
<!-- CONTEXT: To be used in any <xs:complexType>-->
<xs:attribute name="acyclic" type="xs:boolean">
  <xs:annotation>
    <xs:documentation>
      If this attribute is set to true for a type D
      then instances of D should not create any
      cycles in a model. See Section 5.1.1.3 titled "Instance Validity
Rules".
    </xs:documentation>

```



```

</xs:annotation>
</xs:attribute>
<!-- CONTEXT: To be used in <sch:assert>, <sch:report>
and elements with textual content.
This attribute is used to support string localization.
It is used to define the translation location for
the text content of the containing element.-->
<xs:attribute name="locid" type="xs:QName"/>
<!-- CONTEXT: Represents a reference using the URI scheme. To be
used as a child element of elements for which
sml:ref="true". -->
<xs:element name="uri" type="xs:anyURI">
  <xs:annotation>
    <xs:documentation>
      References in URI scheme must be representend by this
      element.
    </xs:documentation>
  </xs:annotation>
</xs:element>
<!--
  Uniqueness and Key constraints
  =====
-->
<xs:complexType name="keybase" mixed="false">
  <xs:sequence minOccurs="0">
    <xs:element
      name="selector"
      type="sml:selectorXPathType"/>
    <xs:element
      name="field"
      type="sml:fieldXPathType"
      maxOccurs="unbounded"/>
    <xs:any
      namespace="##other"
      minOccurs="0"
      maxOccurs="unbounded"
      processContents="lax"/>
  </xs:sequence>
  <xs:attribute name="name" type="xs:NCName"/>
  <xs:attribute name="ref" type="xs:QName"/>
  <xs:anyAttribute
    namespace="##other"
    processContents="lax"/>
</xs:complexType>
<xs:element name="key" type="sml:keybase"/>
<xs:element name="unique" type="sml:keybase"/>
<xs:element name="keyref">
  <xs:complexType mixed="false">
    <xs:complexContent>
      <xs:extension base="sml:keybase">
        <xs:attribute
          name="refer"
          type="xs:QName"
          use="optional"/>
      </xs:extension>
    </xs:complexContent>
  </xs:complexType>
</xs:element>
<!--
  Other Complex Types
  =====

```



```

-->
<xs:complexType name="selectorXPathType" mixed="false">
  <xs:sequence>
    <xs:any
      namespace="##other"
      minOccurs="0"
      maxOccurs="unbounded"
      processContents="lax"/>
  </xs:sequence>
  <xs:attribute name="xpath" use="required">
    <xs:simpleType>
      <xs:restriction base="xs:string">
<!-- The value MUST conform to the selector BNF grammar defined in
      section '4.4 Identity Constraints' in the SML
specification.
-->
</xs:restriction>
    </xs:simpleType>
  </xs:attribute>
  <xs:anyAttribute
    namespace="##other"
    processContents="lax"/>
</xs:complexType>
<xs:complexType name="fieldXPathType" mixed="false">
  <xs:sequence>
    <xs:any
      namespace="##other"
      minOccurs="0"
      maxOccurs="unbounded"
      processContents="lax"/>
  </xs:sequence>
  <xs:attribute name="xpath" use="required">
    <xs:simpleType>
      <xs:restriction base="xs:string">
<!-- The value MUST conform to the field BNF grammar defined in
      section '4.4 Identity Constraints' in the SML
specification.
-->
</xs:restriction>
    </xs:simpleType>
  </xs:attribute>
  <xs:anyAttribute
    namespace="##other"
    processContents="lax"/>
</xs:complexType>
</xs:schema>

```

## В. Приклад визначення моделі документа (ненормативний)

Даний приклад [визначення моделі документа](#) ілюструє використання наступних розширень SML:

1. Посилання MCM
2. Ключові обмеження та обмеження ключових посилань
3. Обмеження схематрона

```

<?xml version="1.0" encoding="utf-8"?>

<!--
/*
 * Copyright © World Wide Web Consortium,
 *
 * (Massachusetts Institute of Technology, European Research Consortium for
 * Informatics and Mathematics, Keio University). All Rights Reserved. This
 * work is distributed under the W3C® Document License [1] in the hope that
 * it will be useful, but WITHOUT ANY WARRANTY; without even the implied
 * warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.
 *
 * [1] http://www.w3.org/Consortium/Legal/2002/copyright-documents-20021231
 */
-->
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema"
xmlns:tns="http://example.org/SampleModel" xmlns:sml="http://www.w3.org/ns/sml"
xmlns:smlfn="http://www.w3.org/ns/sml-function"
xmlns:sch="http://purl.oclc.org/dsdl/schematron"
targetNamespace="http://example.org/SampleModel" elementFormDefault="qualified"
finalDefault="" blockDefault="" attributeFormDefault="unqualified">

  <xs:simpleType name="SecurityLevel">
    <xs:restriction base="xs:string">
      <xs:enumeration value="Low"/>
      <xs:enumeration value="Medium"/>
      <xs:enumeration value="High"/>
    </xs:restriction>
  </xs:simpleType>

  <xs:complexType name="Hostref" sml:acyclic="true" mixed="false">
    <xs:sequence>
      <xs:any namespace="##any" processContents="lax" minOccurs="0"
maxOccurs="unbounded"/>
    </xs:sequence>
    <xs:anyAttribute namespace="##any" processContents="lax"/>
  </xs:complexType>

  <!-- This element represents the host operating system for
an application. Note that the type of the referenced
element must be OperatingSystemType or a derived type
of OperatingSystemType -->

  <xs:element name="HostOSRef" type="tns:Hostref"
sml:targetType="tns:OperatingSystemType"/>

  <xs:complexType name="ApplicationType" mixed="false">
    <xs:sequence>
      <xs:element name="Name" type="xs:string"/>
      <xs:element name="Vendor" type="xs:string"/>
      <xs:element name="Version" type="xs:string"/>
      <xs:element ref="tns:HostOSRef" minOccurs="0"/>
    </xs:sequence>
  </xs:complexType>

  <xs:simpleType name="ProtocolType">
    <xs:list>
      <xs:simpleType>
        <xs:restriction base="xs:string">
          <xs:enumeration value="TCP"/>
        </xs:restriction>
      </xs:simpleType>
    </xs:list>
  </xs:simpleType>

```

```

        <xs:enumeration value="UDP"/>
        <xs:enumeration value="SMTP"/>
        <xs:enumeration value="SNMP"/>
    </xs:restriction>
</xs:simpleType>
</xs:list>
</xs:simpleType>

<xs:complexType name="GuestAppRefType" sml:acyclic="false" mixed="false">
    <xs:sequence>
        <xs:any namespace="##any" processContents="lax" minOccurs="0"
maxOccurs="unbounded"/>
    </xs:sequence>
    <xs:anyAttribute namespace="##any" processContents="lax"/>
</xs:complexType>

<xs:element name="GuestAppRef" type="tns:GuestAppRefType"
sml:targetType="tns:ApplicationType"/>

<xs:complexType name="OperatingSystemType" mixed="false">
    <xs:sequence>
        <xs:element name="Name" type="xs:string"/>
        <xs:element name="FirewallEnabled" type="xs:boolean"/>
        <xs:element name="Protocol" type="tns:ProtocolType"/>
    </xs:sequence>
    <!-- The following element represents the applications hosted by
operating system -->
    <xs:element name="Applications" minOccurs="0">
        <xs:complexType mixed="false">
            <xs:sequence>
                <xs:element ref="tns:GuestAppRef" maxOccurs="unbounded"/>
            </xs:sequence>
        </xs:complexType>
    </xs:element>
</xs:sequence>
</xs:complexType>

<xs:complexType name="OSRefType" sml:acyclic="false" mixed="false">
    <xs:sequence>
        <xs:any namespace="##any" processContents="lax" minOccurs="0"
maxOccurs="unbounded"/>
    </xs:sequence>
    <xs:anyAttribute namespace="##any" processContents="lax"/>
</xs:complexType>

<xs:element name="OSRef" type="tns:OSRefType"
sml:targetType="tns:OperatingSystemType"/>

<xs:complexType name="WorkstationType" mixed="false">
    <xs:sequence>
        <xs:element name="Name" type="xs:string"/>
        <xs:element ref="tns:OSRef"/>
        <xs:element name="Applications" minOccurs="0">
            <xs:complexType mixed="false">
                <xs:sequence>
                    <xs:element ref="tns:GuestAppRef" maxOccurs="unbounded"/>
                </xs:sequence>
            </xs:complexType>
        </xs:element>
    </xs:sequence>
</xs:complexType>

```

```

</xs:complexType>

<xs:element name="Workstation" type="tns:WorkstationType">
  <xs:annotation>
    <xs:appinfo>
      <sch:schema>
        <sch:ns prefix="sm" uri="SampleModel"/>
        <sch:ns prefix="smlfn" uri="http://www.w3.org/ns/sml-function"/>
        <sch:pattern id="OneHostOS">

          <!-- The constraints in the following rule are evaluated
                For all instances of the Workstation global element-->

            <sch:rule context=".">

              <!-- define a named variable - MyApplications -
                    for use in test expression-->

                <sch:let name="MyApplications"
value="smlfn:deref(sm:Applications/sm:GuestAppRef)"/>
                <sch:assert
test="count($MyApplications)=count($MyApplications/sm:HostOSRef)">
                  Each application in workstation
                  <sch:value-of select="string(sm:Name)"/>
                  must be hosted on an operating system
                </sch:assert>
              </sch:rule>
            </sch:pattern>
          </sch:schema>

          <!-- In a workstation, (Vendor,Name,Version) is the key for
                guest applications -->

            <sml:key name="GuestApplicationKey">
              <sml:selector xpath="smlfn:deref(tns:Applications/tns:GuestAppRef)"/>
              <sml:field xpath="tns:Vendor"/>
              <sml:field xpath="tns:Name"/>
              <sml:field xpath="tns:Version"/>
            </sml:key>

          <!-- In a workstation, Name is the key for operating system -->

            <sml:key name="OSKey">
              <sml:selector xpath="smlfn:deref(tns:OSRef)"/>
              <sml:field xpath="tns:Name"/>
            </sml:key>

          <!-- In a workstation, the applications hosted by the
                referenced operatinsystem must be a subset of the
                applications in the workstation -->

            <sml:keyref name="OSGuestApplication" refer="tns:GuestApplicationKey">
              <sml:selector
xpath="smlfn:deref(tns:OSRef)/tns:Applications/tns:GuestAppRef"/>
              <sml:field xpath="tns:Vendor"/>
              <sml:field xpath="tns:Name"/>
              <sml:field xpath="tns:Version"/>
            </sml:keyref>

```

```

    <!-- In a workstation, the host operating system of guest
         applications must be a subset of the operating system in
         the workstation -->

    <sml:keyref name="ApplicationHostOS" refer="tns:OSKey">
      <sml:selector
xpath="smlfn:deref(tns:Applications/tns:GuestAppRef)/tns:HostOSRef"/>
      <sml:field xpath="tns:Name"/>
    </sml:keyref>
  </xs:appinfo>
</xs:annotation>
</xs:element>

<xs:element name="SecureWorkstation" type="tns:WorkstationType">
  <xs:annotation>
    <xs:appinfo>
      <sch:schema>
        <sch:ns prefix="sm" uri="SampleModel"/>
        <sch:ns prefix="smlfn" uri="http://www.w3.org/ns/sml-function"/>
        <sch:pattern id="SecureApplication">
          <sch:rule context="sm:Applications/sm:Application">
            <sch:report test="smlfn:deref(.)[sm:SecurityLevel!='High']">
              Application <sch:value-of select="string(sm:Name)"/>
              from <sch:value-of select="string(sm:Vendor)"/>
              does not have high security level.
            </sch:report>
            <sch:assert test="smlfn:deref(.)[sm:Vendor='TrustedVendor']">
              A secure workstation can only contain
              applications from TrustedVendor.
            </sch:assert>
          </sch:rule>
        </sch:pattern>
      </sch:schema>
    </xs:appinfo>
  </xs:annotation>
</xs:element>
</xs:schema>

```

## С. Приклад посилання МСМ (ненормативний)

Наступний приклад ілюструє використання посилань МСМ. Наведемо фрагмент схеми:

```

<xs:element name="EnrolledCourse">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="Name" type="xs:string"/>
      <xs:element name="Grade" type="xs:string"/>
      <xs:any namespace="##any" minOccurs="0"
        maxOccurs="unbounded" processContents="lax"/>
    </xs:sequence>
    <xs:anyAttribute namespace="##any" processContents="lax"/>
  </xs:complexType>
</xs:element>

<xs:complexType name="StudentType">
  <xs:sequence>
    <xs:element name="ID" type="xs:string"/>

```

```

<xs:element name="Name" type="xs:string"/>
<xs:element name="EnrolledCourses" minOccurs="0">
  <xs:complexType>
    <xs:sequence>
      <xs:element ref="tns:EnrolledCourse" maxOccurs="unbounded"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
</xs:sequence>
</xs:complexType>

```

Сема визначення в наведеному вище прикладі являє собою агностику МСМ і не використовує будь-які атрибути МСМ, елементи або типи. Однак елемент `EnrolledCourse` має відкритий конвент моделі і це може бути використано, щоб відзначити приклади `EnrolledCourse` як посилання МСМ, як показано нижче:

```

<Student xmlns="http://www.university.example.org/ns"
  xmlns:sml="http://www.w3.org/ns/sml"
  xmlns:u="http://www.university.example.org/ns">
  <ID>1000</ID>
  <Name>John Doe</Name>
  <EnrolledCourses>
    <EnrolledCourse sml:ref="true">
      <Name>PHY101</Name>
      <Grade>A</Grade>
      <sml:uri>
        http://www.university.example.org/Universities/MIT/Courses.xml
        #smlxpath1(/u:Courses/u:Course[u:Name='PHY101'])
      </sml:uri>
    </EnrolledCourse>
    <EnrolledCourse sml:ref="false">
      <Name>MAT100</Name>
      <Grade>B</Grade>
      <sml:uri>
        http://www.university.example.org/Universities/MIT/Courses.xml
        #smlxpath1(/u:Courses/u:Course[u:Name='MAT100'])
      </sml:uri>
    </EnrolledCourse>
    <EnrolledCourse>
      <Name>SocialSkills</Name>
      <Grade>F</Grade>
    </EnrolledCourse>
  </EnrolledCourses>
</Student>

```

Перший елемент `EnrolledCourse` в наведеному вище прикладі являє собою посилання МСМ, оскільки він вказує, що `sml:ref="true"`. Він використовує URI посилання схеми МСМ для визначення цільового елементу `PHY101`. Другий і третій `EnrolledCourse` елементи не являються посиланнями МСМ; другий елемент вказує, що `sml:ref="false"`, і третій елемент не вказує `sml:ref` атрибут. Зверніть увагу, що другий `EnrolledCourse` елемент містить `sml:uri` елемент, який задовольняє синтаксис URI посилання схеми МСМ (мається на увазі звичайно `MAT100`), але це буде проігноровано, оскільки `sml:ref="false"` для цього `EnrolledCourse` елементу.

Зверніть увагу, що немає ніяких обмежень МСМ, визначених на `EnrolledCourse` елементі або на типі цього елемента в схемі. Тому, навіть якщо перший приклад `EnrolledCourse` елементу позначений як посилання МСМ, жодне обмеження МСМ не оцінюються для

цього елемента під час перевірки моделі. Тим не менш, перевірка таких, як визначено в розділі [4.2.1 не більше однієї мети](#) і в розділі [4.2.2 послідовні довідки](#), як і раніше здійснюється на цих посиланнях MCM в ході перевірки моделі.

EnrolledCourse посилання MCM можуть бути позначені як нульові посилання, якщо вони визначають `sml:nilref="true"` атрибут, як показано в наступному прикладі (перший елемент EnrolledCourse є нульовим посиланням MCM):

```
<Student xmlns="http://www.university.example.org/ns"
  xmlns:sml="http://www.w3.org/ns/sml"
  xmlns:u="http://www.university.example.org/ns">
  <ID>1000</ID>
  <Name>John Doe</Name>
  <EnrolledCourses>
    <EnrolledCourse sml:ref="true" sml:nilref="true">
      <Name>PHY101</Name>
      <Grade>A</Grade>
    </EnrolledCourse>
    <EnrolledCourse sml:ref="false">
      <Name>MAT100</Name>
      <Grade>B</Grade>
      <sml:uri>
        http://www.university.example.org/Universities/MIT/Courses.xml
        #smlxpath1(/u:Courses/u:Course[u:Name='MAT100'])
      </sml:uri>
    </EnrolledCourse>
    <EnrolledCourse>
      <Name>SocialSkills</Name>
      <Grade>F</Grade>
    </EnrolledCourse>
  </EnrolledCourses>
</Student>
```

У наведеному вище прикладі перше посилання MCM, EnrolledCourse, визначає `sml:nilref="true"` атрибут, який відзначений як нульове посиланням MCM. Вказавши нульове посилання, автор документа робить відповідну заяву, що цей елемент `Student` не відноситься до будь-якої мети елемента. Вказівка нульового посилання не має жодного впливу, визначеного MCM, на інтерпретацію елементів у контекстах не MCM. Зокрема, в даному випадку, MCM нічого не говорить про тлумачення `Grade` та `Name` елемента. Будь-яке таке тлумачення залишено на застосування, його використовує контекст інших специфікацій і т.д.

## D. Приклад URI посилання схеми MCM (ненормативний)

Наступний приклад ілюструє використання URI посилання схеми MCM [4.3.1 URI посилання схеми MCM]. Розглянемо випадок, коли всі курси, запропоновані MIT зберігаються в одному документі XML - `Courses.xml` - чиє URI є `http://www.university.example.org/Universities/MIT/Courses.xml`. У цьому випадку, елемент всередині `Courses.xml` що відповідає курсу PHY101, може бути представлений наступним чином (при умові, що `Coursesis` - кореневий елемент в

`Courses.xml`)

```
<Student xmlns="http://www.university.example.org/ns">
  <ID>1000</ID>
```

```

<Name>John Doe</Name>
<EnrolledCourses>
  <EnrolledCourse sml:ref="true"
xmlns:u="http://www.university.example.org/ns">
  <sml:uri>
    http://www.university.example.org/Universities/MIT/Courses.xml
    #smlxpath1(/u:Courses/u:Course[u:Name='PHY101'])
  </sml:uri>
</EnrolledCourse>
</EnrolledCourses>
</Student>

```

Посилання MCM може також звертатися до елементу в своєму документі. Щоб побачити це, розглянемо наступний приклад документу

```

<University xmlns="http://www.university.example.org/ns">
  <Name>MIT</Name>
  <Courses>
    <Course>
      <Name>PHY101</Name>
    </Course>
    <Course>
      <Name>MAT200</Name>
    </Course>
  </Courses>
  <Students>
    <Student>
      <ID>123</ID>
      <Name>Jane Doe</Name>
      <EnrolledCourses>
        <EnrolledCourse sml:ref="true"
xmlns:u="http://www.university.example.org/ns">
          <sml:uri>
            #smlxpath1(/u:University/u:Courses/u:Course[u:Name='MAT200'])
          </sml:uri>
        </EnrolledCourse>
      </EnrolledCourses>
    </Student>
  </Students>
</University>

```

Тут елемент EnrolledCourse для посилання student Jane Doe є елементом Course для елементу MAT200 в цьому ж документі.

## Е. Приклад обмеження ідентичності MCM (ненормативний)

У наступному прикладі будуть використовуватися для ілюстрації `sml:key`, `sml:unique`, та `sml:keyref` обмеження посилань MCM. Цей приклад складається з трьох схематичних документів. `university.xsd` містить визначення типів для елементу `University`, посилання MCM `Student` і посилання MCM `Course`. `students.xsd` містить визначення типів для посилання `EnrolledCourse` MCM і елементу `Student`. `courses.xsd` містить визначення типу для елементу `Course`.

```

<!-- from university.xsd -->
<xs:complexType name="StudentRefType">
  <!-- SML reference to a Student -->

```



```

    <xs:sequence>
      <xs:any namespace="##any" processContents="lax" minOccurs="0"
maxOccurs="unbounded" />
    </xs:sequence>
    <xs:anyAttribute namespace="##any" processContents="lax" />
  </xs:complexType>

  <xs:element name="Student" type="StudentRefType" />

  <xs:complexType name="CourseRefType">
    <!-- SML reference to a Course -->
    <xs:sequence>
      <xs:any namespace="##any" processContents="lax" minOccurs="0"
maxOccurs="unbounded" />
    </xs:sequence>
    <xs:anyAttribute namespace="##any" processContents="lax" />
  </xs:complexType>

  <xs:element name="Course" type="CourseRefType" />

  <xs:complexType name="UniversityType">
    <xs:sequence>
      <xs:element name="Name" type="xs:string" />
      <xs:element name="Students" minOccurs="0">
        <xs:complexType>
          <xs:sequence>
            <xs:element ref="Student" maxOccurs="unbounded" />
          </xs:sequence>
        </xs:complexType>
      </xs:element>
      <xs:element name="Courses" minOccurs="0">
        <xs:complexType>
          <xs:sequence>
            <xs:element ref="Course" maxOccurs="unbounded" />
          </xs:sequence>
        </xs:complexType>
      </xs:element>
    </xs:sequence>
  </xs:complexType>

  <!-- from students.xsd -->
  <xs:complexType name="EnrolledCourseRefType">
    <!-- SML reference to a Course -->
    <xs:sequence>
      <xs:element name="Grade" type="xs:string" />
      <xs:any namespace="##any" processContents="lax" minOccurs="0"
maxOccurs="unbounded" />
    </xs:sequence>
    <xs:anyAttribute namespace="##any" processContents="lax" />
  </xs:complexType>

  <xs:element name="EnrolledCourse" type="EnrolledCourseRefType" />

  <xs:complexType name="StudentType">
    <xs:sequence>
      <xs:element name="ID" type="xs:string" />
      <xs:element name="SSN" type="xs:string" minOccurs="0" />
      <xs:element name="Name" type="xs:string" />
      <xs:element name="EnrolledCourses" minOccurs="0">
        <xs:complexType>
          <xs:sequence>

```

```

        <xs:element ref="EnrolledCourse" maxOccurs="unbounded"/>
    </xs:sequence>
</xs:complexType>
</xs:element>
</xs:sequence>
</xs:complexType>

<xs:element name="Students">
    <xs:complexType>
        <xs:sequence>
            <xs:element name="Student" type="StudentType"/>
        </xs:sequence>
    </xs:complexType>
</xs:element>

<!-- from courses.xsd -->
<xs:complexType name="CourseType">
    <xs:sequence>
        <xs:element name="Name" type="xs:string"/>
        <xs:element name="EnrolledStudents" minOccurs="0">
            <xs:complexType>
                <xs:sequence>
                    <xs:element name="EnrolledStudent" maxOccurs="unbounded">
                        <xs:complexType>
                            <xs:sequence>
                                <xs:element name="StudentID" type="xs:string"/>
                            </xs:sequence>
                        </xs:complexType>
                    </xs:element>
                </xs:sequence>
            </xs:complexType>
        </xs:element>
    </xs:sequence>
</xs:complexType>
</xs:element>
</xs:sequence>
</xs:complexType>

<xs:element name="Courses">
    <xs:complexType>
        <xs:sequence>
            <xs:element name="Course" type="CourseType"/>
        </xs:sequence>
    </xs:complexType>
</xs:element>

```

## sml:key and sml:unique

XML схема підтримує ключові і унікальні обмеження через `xs:key` та `xs:unique`, але ці обмеження можуть бути визначені тільки в рамках одного документа XML. `sml:key` and `sml:unique` елементи підтримують специфікацію ключових і унікальних обмежень документа. Будемо використовувати визначення `UniversityType`, щоб проілюструвати цю концепцію. Логічно очікувати, що кожен студент в університеті повинен мати унікальну самотутність, і ця ідентичність повинна бути вказана. Це може бути виражена таким чином:

```

<xs:element name="University" type="tns:UniversityType">
    <xs:annotation>
        <xs:appinfo>
            <sml:key name="StudentIDisKey">
                <sml:selector
xpath="smlfn:deref(tns:Students/tns:Student)/tns:ID"/>
                <sml:field xpath="."/>
            </sml:key>
        </xs:appinfo>
    </xs:annotation>
</xs:element>

```

```
        </sml:key>
    </xs:appinfo>
</xs:annotation>
</xs:element>
```

`sml:key` та `sml:unique` обмеження аналогічні, але не ті ж самі. `sml:key` вимагає, щоб вказане поле повинне бути присутнім у прикладі документа та мати унікальні значення, у той час як `sml:unique` просто вимагає певні області мати унікальні значення, але не потрібна їх присутність у прикладі документа. Таким чином ключі мають на увазі унікальність, а унікальність не має на увазі ключі. Наприклад, студенти, які вчаться в університеті повинні мати унікальний номер соціального страхування, але університет може ще налічувати іноземних студентів, які не володіють цим номером. Це обмеження може бути вказане наступним чином:

```
<xs:element name="University" type="tns:UniversityType">
  <xs:annotation>
    <xs:appinfo>
      <sml:unique name="StudentSSNisUnique">
        <sml:selector xpath="smlfn:deref(tns:Students/tns:Student)"/>
        <sml:field xpath="tns:SSN"/>
      </sml:unique>
    </xs:appinfo>
  </xs:annotation>
</xs:element>
```

`sml:key` та `sml:unique` обмеження завжди зазначені в контексті узагальнюючого елемента. У наведеному вище прикладі декларація елемента `University` є контекстом для ключових і унікальних обмежень.

Наступний приклад ілюструє використання `ref` атрибуту в обмеженні ідентичності MCM:

```
<xs:element name="PrivateUniversity" type="tns:UniversityType">
  <xs:annotation>
    <xs:appinfo>
      <sml:unique ref="tns:StudentSSNisUnique"/>
    </xs:appinfo>
  </xs:annotation>
</xs:element>
```

У наведеному вище прикладі декларація елемента `PrivateUniversity` вказує на унікальне обмеження `StudentSSNisUnique` за посиланням його назви в декларації елемента `declaration`.

## SML: keyref

XML схема підтримує ключові посилання через `xs:keyref`, щоб один набір значень був підмножиною іншого набору значень в документі XML. Такі обмеження є аналогічними з іноземними ключами в реляційних базах даних. Основні посилання в XML схемі підтримується тільки в рамках одного документа XML. `xs:keyref` елемент дозволяє ключовим посиланням бути вказаними через посилання MCM і через XML документи. Наступний приклад використовує `xs:keyref`, щоб захопити потребу, що учні звичайно ж повинні бути зараховані в даний час в університеті:

```
<xs:element name="University" type="tns:UniversityType">
```

```

<xs:annotation>
  <xs:appinfo>
    <sml:key name="StudentIDisKey">
      <sml:selector xpath="smlfn:deref(tns:Students/tns:Student)"/>
      <sml:field xpath="tns:ID"/>
    </sml:key>
    <sml:keyref name="CourseStudents" refer="tns:StudentIDisKey">
      <sml:selector
xpath="smlfn:deref(tns:Courses/tns:Course)/tns:EnrolledStudents/tns:EnrolledStude
nt"/>
      <sml:field xpath="tns:ID"/>
    </sml:keyref>
  </xs:appinfo>
</xs:annotation>
</xs:element>

```

Дане обмеження вказує, що для університету набір ідентифікаторів студентів, зарахованих на курс, є підмножиною набору ідентифікаторів студентів, які вже навчаються в університеті. Зокрема, `selector` та `field` елементи в `StudentIDisKey` ключі являються основною перешкодою для визначення набору ідентифікаторів студентів, які вже навчаються в університеті, а також `selector` та `field` елементи в `StudentIDisKey` ключі посилання являються основною перешкодою визначення набору ідентифікаторів студентів, які навчаються на курсах.

## Ф. Зразки локалізації та підстановки змінних (ненормативні)

Наступні приклади демонструють, як локалізація може застосовуватися для текстових повідомлень в результаті оцінки правила схематрона, дозволяючи [модель процесора](#) підтримувати кілька локалей без змін для моделі процесора або правил схематрона. Нижче коротко викладаються переваги, що впливають з використання `sml:locid` підтримки локалізації:

1. Правило схематрону текстових повідомлень являються локально-незалежним в тому сенсі, в якому автор не повинен мати справу з виконанням локальних правил оцінки. Правило схематрону визначається генетично, видатково для будь-якого експерта з оцінки (у тому числі моделі процесора), для якого файл перекладу стає доступним на місці визначення `sml:locid` значення [назви ім'я простору](#).
2. Існує чіткий поділ між перекладом текстових повідомлень, авторським правилом схематрона та оціночним кодом правила схематрона. Правила схематрона вимагають, щоб жодні зміни не були доступними при перекладах на інші мови. Те ж правило схематрона може бути використане кількома оцінювачами, кожний з яких підтримує індивідуальний набір мови. Для підтримки нової мови все, що потрібно зробити, це додати новий файл перекладу відповідно до місця визначення `sml:locid` значення [назви ім'я простору](#).

Спираючись на попередній приклад університету, наступний приклад правила схематрона використовує `sml:locid` атрибут для пошуку переказу інформації для `sch:assert` схематрона, щоб повідомити про помилку:

```

<sch:schema xmlns:sch="http://purl.oclc.org/dsdl/schematron"
             xmlns:lang="http://www.university.example.org/translation/">

  <sch:ns prefix="u" uri="http://www.university.example.org/ns" />
  <sch:ns prefix="smlfn" uri="http://www.w3.org/ns/sml-function"/>
  <sch:pattern id="StudentPattern">
    <sch:rule context="u:Students/u:Student">
      <sch:assert test=".[starts-with(u:ID,'99')]"
                 sml:locid="lang:StudentIDErrorMsg">
        The specified ID <sch:value-of select="string(u:ID)"/> does not
begin with 99.
      </sch:assert>
    </sch:rule>
  </sch:pattern>
</sch:schema>

```

Загалом, [назва ім'я простору](#) може вказати файл, що містить перекладені повідомлення, папки, яка містять набір перекладених файлів або будь-які інші види ресурсів, які можуть допомогти знайти перекладені повідомлення. Це залежить від реалізації того, як модель валідатора дозволяє використовувати цю інформацію для знаходження фактичного ресурсу, що містить перекладені повідомлення. У цьому конкретному прикладі [назва ім'я простору](#) значення `sml:locid` атрибута використовується для визначення місцезорозташування ресурсу, що містить перекладений текст:

```
xmlns:lang="http://www.university.example.org/translation/"
```

Крім того, в цьому конкретному прикладі, `http://www.university.example.org/translation/` назви папки, яка містить набір перекладених ресурсів і є один набір файлів перекладу, розташованих на `http://www.university.example.org/translation/`. Кожен з цих файлів перекладу відповідає мові, в якій повідомлення були перекладені. Переклади на французьку і німецьку доступні в наступних файлів:

1. Файл `Http://www.university.example.org/translation/lang_fr.txt` містить французький переклад `sch:assert` повідомлення.
2. Файл `Http://www.university.example.org/translation/lang_de.txt` містить німецький переклад `sch:assert` повідомлення.

{Місцева частин} значення `sch:assert` атрибуту (`StudentIDErrorMsg`) використовується для визначення особи, яка перекладає повідомлення. Ця тотожність використовується для виявлення перекладеного тексту в перекладі ресурсів.

Французький переклад `sch:assert` повідомлень знаходиться, як подано нижче:

```
StudentIDErrorMsg = L'identifiant specifiy <sch:value-of
select="string(u:ID)"/> ne commence pas par 99.
```

Німецький переклад для `sch:assert` повідомлень знаходиться, як подано нижче:

```
StudentIDErrorMsg = Der angegebene Wert des Attributs ID (<sch:value-
of select="string(u:ID)"/>) beginnt nicht mit "99".
```

Повідомлення, які перекладаються, особливо рядки, що містять XML мітки (такі, як в прикладі нижче `<sch:value-of select="string(u:ID)"/>`), можуть бути найкращими зберігачами в контейнерах XML. Це дозволяє більш гнучко маніпулювати та перекладати дані. Наприклад, документ XML може використовувати його, щоб додати локалізовану інформацію.

```
<?xml version="1.0" encoding="UTF-8"/>
<messages xml:lang="en"
  xmlns:sch="http://purl.oclc.org/dsdl/schematron"
  xmlns:its="http://www.w3.org/2005/11/its">
  <msg xml:id='StudentIDErrorMsg'
    its:locNote="This message should not be longer than 128 characters">
    The specified ID <sch:value-of select="string(u:ID)"/> does not begin
with 99.
  </msg>
</messages>
```

### Найкраща практична рекомендація для підстановки змінної

Часто буває, що перекладений текст може бути повторно використаний, наприклад, повідомлення схематрона можна повторно використовувати в різних `sch:assert` або `sch:report` правилах та закономірностях. У наведеному вище прикладі автор правила схематрона може використовувати це повідомлення в іншому контексті:

```
The specified ID <sch:value-of select="string(u:ID)"/> does not begin with 99.
```

Повторне використання можливостей для цього повідомлення є обмеженим, оскільки повідомлення, яке перекладається, містить припущення про правила контексту. Щоб мати можливість повторно використовувати дане повідомлення більш широко, повідомлення автора має замінити контекстно-незалежний вираз, такий як змінна, замість `u:ID` в `<sch:value-of select="string(u:ID)"/>`.

Таким чином, оригінальний переклад повідомлення:

```
StudentIDErrorMsg = L'identifiant spécifique <sch:value-of select="string(u:ID)"/>
ne commence pas par 99.
StudentIDErrorMsg = Der angegebene Wert des Attributs ID (<sch:value-of
select="string(u:ID)"/>) beginnt nicht mit "99".
```

Замість цього слід використовувати кодовані значення змінної, як показано нижче.

```
StudentIDErrorMsg = L'identifiant spécifique <sch:value-of select="$studentID"/> ne
commence pas par 99.
StudentIDErrorMsg = Der angegebene Wert des Attributs ID (<sch:value-of
select="$studentID"/>) beginnt nicht mit "99".
```

Помилкове повідомлення в `sch:assert`, виявлене значенням `lang:StudentIDErrorMsg`, може бути використане в еконтекстах правила схематрона, окрім випадків, описаних вище. У випадках, коли підстановка змінних використовуються таким чином, відповідальність за встановлення змінної в різних контекстах повідомлення лежить на верховенстві автора (ів). Зокрема, правила схематрона використовуються на початку Додатку F, для запуску цього прикладу необхідно встановити значення `studentID` змінної. Нижче наведений приклад показує, як переглянуті ресурси перекладу будуть

посилатися з правил схематрона.

```
<sch:schema xmlns:sch="http://purl.oclc.org/dsdl/schematron"
  xmlns:lang="http://www.university.example.org/translation/">
  <sch:ns prefix="u" uri="http://www.university.example.org/ns" />
  <sch:ns prefix="smlfn" uri="http://www.w3.org/ns/sml-function" />
  <sch:pattern id="StudentPattern">
    <sch:rule context="u:Students/u:Student">
      <sch:let name="studentID" value="u:ID"/>
      <sch:assert test="starts-with(u:ID,'99') "
        sml:locid="lang:StudentIDErrorMsg">
        The specified ID <sch:value-of select="$studentID"/> does not
begin with 99.
      </sch:assert>
    </sch:rule>
  </sch:pattern>
</sch:schema>
```

The error message in `sch:assert` and the localization identifier `lang:StudentIDErrorMsg` can now be reused in contexts other than `u:Students/u:Student`.

Якщо перекладений рядок, названий `StudentIDErrorMsg`, призначений для використання тільки в контексті елементів Schematron, цей механізм є достатнім. Якщо повідомлення призначене для використання в додаткових контекстах, попереднього синтаксису не завжди достатньо. З метою забезпечення повторного використання перекладених рядків, експлуатувавши підстановку змінних в інших контекстах, розуміння синтаксису обома: внутрішнім або зовнішнім, елементами схематрона не потрібне. `xsl:variable` та `sch:let` - можливі випадки, вирбляють більш практичні вимоги МСМ, такі, при яких валідатори підтримки "XSLT" запитів обов'язкові для схематрона. Однак, в даний час недостатнім є практичний досвід для позначення цього передової практики.

## Г. Подяка (ненормативна)

Редактори визнають членів робочої групи Мови Службового Моделювання, членів інших W3C робочих груп, а також галузевих експертів в рамках інших форумів, які внесли свій внесок, прямо або побічно пов'язаних для процесу створення, або змісту цього документа.

На момент публікації цієї специфікації членами робочої групи Мови Службового Моделювання були:

Джон Арве (Корпорація IBM), Льон Шар (Корпорація Microsoft), Сенді Гао (Корпорація IBM), Поль Ліптон (CA), Джеймс Лін (HP), Кумар Пандіт (Корпорація Microsoft), Валентин Попеску (Корпорація IBM), Вірджинія Сміт (HP), Генрі Томпсон (W3C/ERCIM), Девід Увайтмен (Корпорація IBM), Кірк Уілсон (Каліфорнія).

Робоча група Мови Службового Моделювання користується у своїй роботі участю та вкладом числа людей, які в даний час не є членами Робочої групи, включаючи, зокрема, тих, хто зазначен нижче.

Дейв Енебуске (IBM), Джон Хасс (Dell), Стів Джерман (Cisco), Хізер Крегер (IBM), Вінсент Ковальські (VMC), Мілан Міленкович (Intel), Брайан Мюррей (HP), Філ Прайсек (HP), Джунаїд Сейвд (EMC), Г. Слюйман (IBM), К. Майкл Сперберг-Маккуїн (W3C/MIT),

Бассам Таббара (Microsoft), Віджай Теварі (Intel), Вільям Вамбенеїп (HP), Марв Вашке (CA), Андреа Вестерінен (Microsoft), Пратул Дубліш (Microsoft), Джулія МакКарті (IBM).

Зв'язки, наведені вище, актуальні на момент їх роботи з робочою групою.