

SKOS Проста система організації знань

Довідкова інформація

Рекомендація W3C від 18 серпня 2009

Ця версія:

<http://www.w3.org/TR/2009/REC-skos-reference-20090818/>

Остання версія:

<http://www.w3.org/TR/skos-reference>

Попередні версії:

<http://www.w3.org/TR/2009/PR-skos-reference-20090615/>

Редактори:

[Алістер Милі](#), STFC Rutherford Appleton Laboratory / Оксфордський університет

[Шон Беххофер](#), Університет Манчестера

Будь ласка, зверніться до [помилки](#) в цьому документі, де можуть міститися деякі нормативні виправлення.

Див також [переклади](#).

Copyright © 2009 W3C® ([MIT](#), [ERCIM](#), [Keio](#)). Всі права захищені. Застосовуються правила [відповідальності](#), [товарних знаків](#) і [використання документів](#) W3C.

Анотація

Цей документ визначає просту систему організації знань (SKOS), загальну модель даних для обміну та зв'язку систем організації знань через Інтернет.

Багато систем організації знань, таких, як тезауруси, таксономії, класифікації і заголовки систем, мають аналогічні структури, і можуть бути використані в аналогічних додатках. SKOS використовує більшу частину цих аналогій і робить їх явними для обміну даними та технологіями між різними додатками.

Модель даних SKOS забезпечує стандартний шлях низької вартості для портування існуючих систем організації знань у Semantic Web. SKOS також забезпечує легку, інтуїтивно зрозумілу мову для розробки і поширення нових систем організації знань. Вона може використовуватися самостійно, або у поєднанні з офіційними мовами подання знань, такими як Web Ontology language (OWL).

Цей документ є нормативною специфікацією простої системи організації знань. Він призначений для читачів, які залучені в розробку і впровадження інформаційних систем, які вже мають хороше розуміння технології Semantic Web, особливо RDF і OWL.

Для інформативного керівництва з використання SKOS див. [\[SKOS-PRIMER\]](#).

Підсумок

Використання SKOS: **концепти** можуть бути визначені за допомогою використання URI, **помічені** за допомогою лексичних рядків однією або більше мовами, присвоєні **позначенням** (лексичні коди), **задокументовані** різними типами нотаток, **пов'язані з іншими концептами** і організовані в неофіційні ієрархії та асоціативні мережі, зібрані в **концептуальні схеми**, згруповані в позначених і / або впорядкованих **колекціях**, а також **співставлені** з концептами в інших схемах.

Статус цього документа

Цей розділ описує статус цього документа на момент його публікації. Інші документи можуть замінити цей документ. Список поточних публікацій W3C і остання ревізія цієї технічної доповіді знаходиться в [W3C technical reports index](http://www.w3.org/TR/) на <http://www.w3.org/TR/>.

Даний документ є рекомендацією W3C, розроблений [Semantic Web Deployment Working Group](#), частиною [W3C Semantic Web Activity](#). Цей документ відображає зміни редакційного характеру, що виникають у ході запропонованого огляду рекомендації: ненормативний приклад і попередній текст був видалений, запропоновано один із засобів для посилання до системи позначень (наприклад, символічне позначення) в позначці, де система позначень не відповідає природній мові. Цю пропозицію було визнано несумісною з IETF [Best Current Practice 47](#) з використанням тегів для виявлення мови. Користувач повинен розглянути [SKOS Extension vocabulary](#) для підтримки альтернативних систем позначення. Доповідь про документи відома використанням SKOS в ході звітного періоду кандидатських рекомендацій. Оновлений [SKOS Primer](#) видається одночасно з цією пропозицією.

Зміни після [запропонованої рекомендації 15 червня 2009](#):

- Вилучений останній пункт і приклад з розділу 6.5.4 заснований на приватному використанні мови суб-тегів.
- Редакція внесла зміни до розділу посилання та цитування SKOS Primer.

Коментарі до цього документа можна надсилати на public-swd-wg@w3.org з [громадського архіву](#).

Цей документ був підготовлений групою, що діє в рамках [W3C Patent Policy від 5 лютого 2004](#). W3C підтримує [публічний список будь-яких патентів](#), розроблений у зв'язку з результатами роботи групи; ця сторінка містить інструкції для розкриття патентів. Будь-яка особа, якій фактично було відомо про патент, що містить [основні претензії](#) повинен розкрити інформацію згідно з [розділом 6 W3C Patent Policy](#).

Цей документ був розглянутий членами W3C, розробниками програмного забезпечення, а також іншими групами W3C, і був схвалений директором з якості рекомендацій W3C. Це документ постійний і може бути використаний в якості довідкового матеріалу або цитуватися в інших документах. Роль W3C у розробці даної рекомендації полягає в залученні уваги до цієї специфікації і сприянню її широкому розповсюдженню. Це підвищує функціональність і можливість взаємодії в Інтернеті.

Зміст

- **1. Введення**
 - [1.1. Довідкова інформація та мотивація](#)
 - [1.2. Огляд SKOS](#)
 - [1.3. SKOS, RDF і OWL](#)
 - [1.4. Сумісність і цілісність](#)
 - [1.5. Висновки, залежності і відкритий світ припущень](#)
 - [1.6. Обґрунтування дизайну](#)
 - [1.7. Як читати цей документ](#)
 - [1.7.1. Формальні визначення](#)
 - [1.7.2. Скорочення URI](#)
 - [1.7.3. Приклади](#)
 - [1.8. Відповідність](#)
- **2. Простір імен і словник SKOS**
- **3. Клас skos:Concept**
 - [3.1. Преамбула](#)
 - [3.2. Словник](#)
 - [3.3. Визначення класів & властивостей](#)
 - [3.4. Приклади](#)
 - [3.5. Нотатки](#)
 - [3.5.1. Концепти SKOS, класи OWL і властивості OWL](#)
- **4. Концептуальні схеми**
 - [4.1. Преамбула](#)
 - [4.2. Словник](#)
 - [4.3. Визначення класів & властивостей](#)
 - [4.4. Умови цілісності](#)
 - [4.5. Приклади](#)
 - [4.6. Нотатки](#)
 - [4.6.1. Закриті системи проти відкритих](#)
 - [4.6.2. Концептуальні схеми SKOS і онтології OWL](#)
 - [4.6.3. Головні концепти і семантичні відносини](#)
 - [4.6.4. Схема локалізації та семантичні відносини](#)
 - [4.6.5. Домен skos:inScheme](#)
- **5. Лексичні позначки**
 - [5.1. Преамбула](#)
 - [5.2. Словник](#)
 - [5.3. Визначення класів & властивостей](#)
 - [5.4. Умови цілісності](#)
 - [5.5. Приклади](#)
 - [5.6. Нотатки](#)
 - [5.6.1. Домен властивостей лексичних позначок SKOS](#)
 - [5.6.2. Діапазон властивостей лексичних позначок SKOS](#)
 - [5.6.3. Визначення відносин між позначками](#)
 - [5.6.4. Альтернативи без привілейованих](#)
 - [5.6.5. Позначки і мовні теги](#)
- **6. Позначення**
 - [6.1. Преамбула](#)
 - [6.2. Словник](#)
 - [6.3. Визначення класів & властивостей](#)
 - [6.4. Приклади](#)

- [6.5. Нотатки](#)
 - [6.5.1. Позначення, типізовані літерали і типи даних](#)
 - [6.5.2. Кілька позначень](#)
 - [6.5.3. Унікальні позначення в концептуальних схемах](#)
 - [6.5.4. Позначення і привілейовані позначки](#)
 - [6.5.5. Домен skos:notation](#)
- [7. Властивості документації\(Властивості нотатки\)](#)
 - [7.1. Преамбула](#)
 - [7.2. Словник](#)
 - [7.3. Визначення класів & властивостей](#)
 - [7.4. Приклади](#)
 - [7.5. Нотатки](#)
 - [7.5.1. Домен властивостей документації SKOS](#)
 - [7.5.2. Діапазон властивостей документації SKOS](#)
- [8. Семантичні відносини](#)
 - [8.1. Преамбула](#)
 - [8.2. Словник](#)
 - [8.3. Визначення класів & властивостей](#)
 - [8.4. Умови цілісності](#)
 - [8.5. Приклади](#)
 - [8.6. Нотатки](#)
 - [8.6.1. Відносини під-властивостей](#)
 - [8.6.2. Домен і діапазон властивостей семантичних відносин SKOS](#)
 - [8.6.3. Симетрія skos:related](#)
 - [8.6.4. skos:related і транзитивність](#)
 - [8.6.5. skos:related і рефлексивність](#)
 - [8.6.6. skos:broader і транзитивність](#)
 - [8.6.7. skos:broader і рефлексивність](#)
 - [8.6.8. Цикли в ієрархічних відносинах \(skos:broaderTransitive і рефлексивність\)](#)
 - [8.6.9. Альтернативні шляхи в ієрархічних відносинах](#)
 - [8.6.10. Диз'юнктивність skos:related і skos:broaderTransitive](#)
- [9. Концептуальні колекції](#)
 - [9.1. Преамбула](#)
 - [9.2. Словник](#)
 - [9.3. Визначення класів & властивостей](#)
 - [9.4. Умови цілісності](#)
 - [9.5. Приклади](#)
 - [9.6. Нотатки](#)
 - [9.6.1. Визначені колекції з упорядкованих колекцій](#)
 - [9.6.2. Цілісність skos:memberList](#)
 - [9.6.3. Вкладені колекції](#)
 - [9.6.4. Концепти SKOS, концептуальні колекцій і семантичні відносини](#)
- [10. Властивості співставлення](#)
 - [10.1. Преамбула](#)
 - [10.2. Словник](#)
 - [10.3. Визначення класів & властивостей](#)
 - [10.4. Умови цілісності](#)
 - [10.5. Приклади](#)
 - [10.6. Нотатки](#)
 - [10.6.1. Властивості співставлення, властивості семантичних відносин і концептуальні схеми](#)

- [10.6.2. Розбіжність між ієрархічними і асоціативними зв'язками](#)
- [10.6.3. Властивості співставлення і транзитивність](#)
- [10.6.4. Властивості співставлення і рефлексивність](#)
- [10.6.5. Цикли і альтернативні шляхи, які залучають `skos:broadMatch`](#)
- [10.6.6. Цикли, які залучають `skos:exactMatch` і `skos:closeMatch`](#)
- [10.6.7. Над-властивості рядів, які залучають `skos:exactMatch`](#)
- [10.6.8. `skos:closeMatch`, `skos:exactMatch`, `owl:sameAs`, `owl:equivalentClass`, `owl:equivalentProperty`](#)
- **[11. Список літератури](#)**
- **[12. Подяки](#)**
- **[Додаток А. Властивості і класи SKOS](#)**
 - [А.1. Класи в моделі даних SKOS](#)
 - [А.2. Властивості в моделі даних SKOS](#)
- **[Додаток В. SKOS eXtension for Labels \(SKOS-XL\)](#)**
 - [В.1. Простір імен і словник SKOS-XL](#)
 - [В.2. Клас `skosxl:Label`](#)
 - [В.2.1. Преамбула](#)
 - [В.2.2. Визначення класів і властивостей](#)
 - [В.2.3. Приклади](#)
 - [В.2.4. Нотатки](#)
 - [В.2.4.1. Ідентичність та втілення](#)
 - [В.2.4.2. Склад концептуальних схем](#)
 - [В.3. Привілейовані, альтернативні і приховані `skosxl:Labels`](#)
 - [В.3.1. Преамбула](#)
 - [В.3.2. Визначення класів і властивостей](#)
 - [В.3.3. Приклади](#)
 - [В.3.4. Нотатки](#)
 - [В.3.4.1. *Dumbing-Down* до SKOS лексичних позначок](#)
 - [В.3.4.2. Цілісність позначок SKOS-XL](#)
 - [В.4. Зв'язки між `skosxl:Labels`](#)
 - [В.4.1. Преамбула](#)
 - [В.4.2. Визначення класів і властивостей](#)
 - [В.4.3. Приклади](#)
 - [В.4.4. Нотатки](#)
 - [В.4.4.1. Уточнення цієї схеми](#)
- **[Додаток С. Документи простору імен SKOS і SKOS-XL](#)**
- **[Додаток D. Простір імен SKOS: історичні відомості](#)**

1. Введення

1.1. Довідкова інформація та мотивація

Проста система організації знань - це стандарт обміну даними між різними областями знань, технологій і практики.

В бібліотечних та інформаційних науках, довга і видатна спадщини присвячена розробці інструментів для організації великих колекцій предметів, таких як книги або музей артефактів. Ці інструменти є добре відомі як "система організації знань" (КОС), а інколи, як "контрольовані структуровані словники". Кілька аналогічних, але яскравих традицій виникли з плином часу, кожна з яких підтримується об'єднанням

практиків і встановлює узгоджені стандарти. Різні об'єднання систем організації знань, включаючи тезауруси, класифікаційні схеми, заголовки систем і таксономії широко відомі і застосовуються як в сучасних, так і в традиційних інформаційних систем. На практиці може бути важко провести різницю між абсолютним тезаурусом та класифікаційними схемами або таксономією, хоча деякі властивості можуть бути використані для широкої характеристики цих різних об'єднань (див. наприклад, [\[BS8723-3\]](#)). Важливим моментом для SKOS є те, що на додаток до унікальних особливостей, кожне з цих об'єднань має багато спільного, і найчастіше може бути використане аналогічним чином [\[SKOS-UCR\]](#). Проте в даний час немає широко використовуваного стандарту для подання цих систем організації знань та обміну ними між комп'ютерними системами.

Консорціум W3C Semantic Web [\[SW\]](#) стимулював створення нової області інтеграційних досліджень і розвитку технології на кордонах між системами баз даних, формальною логікою та World Wide Web. Ця робота привела до розробки основоположних стандартів для Semantic Web. Resource Description Framework (RDF) передбачає загальні абстракції даних та синтаксис для Web [\[RDF-PRIMER\]](#). RDF Vocabulary Description language (RDFS) і Web Ontology language (OWL) в сукупності забезпечують загальні дані моделювання (схема) мови для передачі даних в Web [\[RDFS\]](#) [\[OWL-GUIDE\]](#). Мова запитів і протоколів SPARQL забезпечує стандартні засоби для взаємодії з даними в Web [\[SPARQL\]](#).

Ці технології застосовуються в самих різноманітних програмах, оскільки багато програм вимагають загальні рамки для публікації, розповсюдження, обміну та інтеграції ("об'єднання") даних з різних джерел. Можливість пов'язувати дані з різних джерел є мотивацією багатьох проектів, так як різні об'єднання прагнуть використовувати приховані значення даних, раніше поширених ізольованими джерелами.

Один аспект бачення Semantic Web є надія краще організувати величезні обсяги неструктурованої (наприклад, людського розуміння) інформації в Інтернеті, надаючи нові шляхи для відкриття і обміну цією інформацією. RDFS і OWL - формально визначені мови представлення знань, забезпечують способи вираження значення, які піддаються обчисленню, а це доповнює і дає структуру інформації, яка вже присутня в Web [\[RDF-PRIMER\]](#) [\[OWL-GUIDE\]](#). Хоча, щоб реально використовувати ці технології для більш великих обсягів інформації, потрібно побудувати докладну карту окремих галузей знань на додаток до точного опису (наприклад, анотації або каталогізація) інформаційних ресурсів у великих масштабах, більша частина яких не може бути проведена автоматично. Накопичений досвід і кращі практики у галузі бібліотечних та інформаційних наук в області організації інформації і знань, що додатково застосовні до цих концептів, як і багато існуючих систем організації знань є вже розроблені і в експлуатації, такі як Library of Congress Subject Headings [\[LCSH\]](#) або Продовольча та сільськогосподарська організація Організації Об'єднаних Націй AGROVOC Тезаурус [\[AGROVOC\]](#).

Проста система організації знань покликана забезпечити зв'язок між різними громадами практиків в бібліотечних та інформаційних науках, які беруть участь у розробці та застосуванні систем організації знань. Крім того, SKOS націлена на створення зв'язків між цими громадами та Semantic Web шляхом переведення існуючих моделей організації знань в семантичному контексті веб-технологій, а також шляхом забезпечення недорогого шляху для перероблення існуючих систем організації знань в RDF.

Дивлячись в майбутнє, SKOS займає позицію між експлуатацією та аналізом неструктурованої інформації, неформальної і соціально-опосередкованої організації інформації при великому масштабі, і офіційним представлення знань. Доступність накопиченого досвіду організації знань в бібліотечних та інформаційних науках дає можливість застосувати їх до передачі технологічних контекстів Semantic Web таким чином, щоб вони доповнювали існуючі технології Semantic Web (зокрема, формальні системи представлення знань, такі як OWL), і сподіватися, що SKOS приведе до виникнення багатьох нових і цінних програм, а також приведе до нових інтегративних напрямків досліджень і розробок як в технології, так і на практиці.

1.2. Огляд SKOS

Проста система організації знання є спільною моделлю даних для систем організації знань, таких як тезаурус, класифікаційних схем, заголовку систем і таксономії. Використовуючи Skos, система організації знань може бути виражена як **машиночитані дані**. Потім нею можна обмінятися між комп'ютерними програмами та опублікувати в машиночитаній формі в мережі Інтернет.

Модель даних SKOS формально визначена в цій специфікації як онтологія OWL Full [\[OWL-SEMANTICS\]](#). Дані SKOS виражені у вигляді трійки RDF [\[RDF-Concepts\]](#), можуть бути закодовані за допомогою будь-якого конкретного синтаксису RDF (наприклад, RDF / XML [\[RDF-XML\]](#) або Turtle [\[TURTLE\]](#)). Додаткову інформацію про відносини між Skos, RDF і OWL див в наступному підрозділі нижче.

Модель даних SKOS розглядає системи організації знань як **концептуальні схеми** в складі **набору** концептів. Ці концептуальні схеми SKOS та концепти SKOS, що ідентифікуються як URI, дають можливість кожному посилатися на них однозначно в будь-якому контексті і зробити їх частиною World Wide Web. Див [Розділ 3. Клас skos:Concept](#) для отримання більш докладної інформації щодо виявлення та опису концептів SKOS, а [розділ 4. Концептуальні схеми](#) для отримання додаткової інформації про концептуальні схеми.

Концепти SKOS можуть бути **позначені** за допомогою будь-якої кількості лексичних (UNICODE) рядків, таких як "romantic love" або "れんあい", будь-якою природною мовою, наприклад, англійською або японською мовою (написано тут, в хірагана). Одна з цих позначок тією чи іншою мовою може бути зазначена, як привілейована позначка для даної мови, а інші в якості альтернативних позначок. Позначки можуть бути "прихованими", які корисні, коли система організації знань запитала текстовий індекс. Див [розділ 5. Лексичні позначки](#) для отримання додаткової інформації про властивості лексичних позначок SKOS.

Концепти SKOS можуть позначатися одним або декількома **позначеннями**, що є лексичними кодами, які використовуються для однозначної ідентифікації концептів в рамках даної концептуальної схеми. Хоча URI є хорошим засобом виявлення концептів SKOS в комп'ютерних системах, позначення є сполучною ланкою з іншими системами ідентифікацій. Див [Розділ 6. Позначення](#) для отримання більш докладної інформації про позначення.

Концепти SKOS можуть бути **задокументовані** різними типами нотаток. Модель даних SKOS забезпечує базовий набір властивостей документації. Цей набір не є вичерпним, а скоріше забезпечує основу, яка може бути розширена третіми сторонами, щоб забезпечити підтримку для більш конкретних типів нотаток. Див [розділ 7. Властивості документації](#) для отримання більш детальної інформації про нотатки.

Концепти SKOS можуть бути **пов'язані** з іншими концептами SKOS за допомогою властивостей семантичних відносин. Модель даних SKOS забезпечує підтримку ієрархічних і асоціативних зв'язків між концептами SKOS. Знову ж таки, як і в будь-якій частині моделі даних Skos, це може бути розширено третьою стороною, щоб забезпечити підтримку для більш специфічних потреб. Див [розділ 8. Семантичні відносини](#) для отримання більш детальної інформації про зв'язки концептів SKOS.

Концепти SKOS можуть бути згруповані в **колекції**, які можуть бути позначені та / або впорядковані. Ця особливість моделі даних SKOS призначена для забезпечення підтримки позначок в рамках тезаурусу, а також для ситуацій, коли впорядкований набір концептів має сенс або дає деяку корисну інформацію. Див [розділ 9. Концептуальні колекції](#) для отримання більш детальної інформації про колекцію.

Концепти SKOS можуть **бути** співставлені з іншими концептами SKOS в різних концептуальних схемах. Модель даних SKOS забезпечує підтримку чотирьох основних видів зв'язків співставлення: ієрархічні, асоціативні, тісно еквівалентні і точно еквівалентні. Див [розділ 10. Властивості співставлення](#) для отримання більш детальної інформації про співставлення.

Нарешті, факультативне розширення SKOS визначене у [Додатку В. SKOS eXtension for Labels \(SKOS-XL\)](#). SKOS-XL забезпечує більш широку підтримку для виявлення, опису та зв'язку лексичних одиниць.

1.3. Skos, RDF і OWL

Елементами моделі даних SKOS є класи і властивості, а структура і цілісність моделі даних визначається логічними характеристиками і взаємозалежностями між цими класами і властивостями. Це, мабуть, один з найбільш потужних і потенційно суперечливих аспектів Skos, тому що SKOS може бути використаний пліч-о-пліч з OWL, щоб виражати і обмінюватися знаннями про домен, хоча SKOS **не** є офіційною мовою представлення знань.

Щоб зрозуміти цю відмінність, вважають, що "знання", прямо зазначені в офіційній онтології, виражаються як безліч аксіом і фактів. Тезаурус або класифікаційна схема носить зовсім інший характер і не висуває ніяких аксіом чи фактів. Скоріше, тезаурус або класифікаційна схема визначається і описується (природною мовою та іншими неформальними значеннями) безліччю різних ідей і значень, які іноді зручніше називаються "концептами". Ці "концепти" можуть бути також організовані й об'єднані в різні структури, частіше всього в ієрархії і асоціативні мережі. Хоча, ці структури не мають будь-якої формальної семантики і не можуть бути надійно інтерпретовані як формальні аксіомі або факти про навколишній світ. Дійсно, вони ніколи не були призначені для цього, бо вони служать тільки для забезпечення зручної карти деякої предметної області, яка потім може бути використана в якості допомоги для організації та пошуку об'єктів, таких як документи, які мають відношення до цієї галузі.

Для того, щоб організувати "знання" у тезаурус або явну класифікаційну схему потрібно, щоб тезаурус або класифікаційна схема була *реорганізована* в якості офіційної онтології. Іншими словами, деяка особа повинна виконати роботу по перетворенню структури та інтелектуального змісту тезаурусу в набір формальних аксіом і фактів. Ця робота щодо перетворення забирає багато часу, і, отже, є дорогою. Багато чого можна почерпнути з використання тезаурусу і т.д., як є, що є зручним для навігації в рамках предметної області. Використовуючи їх як є, не потребує будь-яких реорганізацій, і тому є набагато дешевшим. Крім того, деякі КОС

не є, за визначенням, призначені для представлення логічних поглядів на домен. Перетворення таких КОС у формальному логічному представленні, можуть на практиці бути пов'язані зі змінами, які призводять до представлення, що не відповідає поставленій меті.

OWL, тим не менше, забезпечена потужною мовою моделювання даних. Отже, ми можемо використовувати OWL для побудови моделі даних для представлення тезаурусів та класифікаційних схем, як є. Це саме те, що робить SKOS. Виходячи з цього підходу, "концепти" тезауруса або класифікаційної схеми моделюються як окремі одиниці в моделі даних Skos, а неофіційні описи і зв'язки між цими "концептами" моделюються як факти про ці одиниці, а не як клас або аксіоми властивостей. Зверніть увагу, що ці *факти* про тезаурус або класифікаційну схему такі, як "концепція X має привілейовану позначку 'Y' і є частиною тезауруса Z"; це **не** факти про те, як світ влаштований у конкретній предметній області, як може бути висловлено в офіційній онтології.

Дані SKOS виражені як трійки RDF. Графа RDF нижче (в [TURTLE](#)) як описано в [розділі 1.7.3](#)) показує деякі факти про тезаурус.

```
<A> rdf:type skos:Concept ;
skos:prefLabel "love"@en ;
skos:altLabel "adoration"@en ;
skos:broader <B> ;
skos:inScheme <S> .
```

```
<B> rdf:type skos:Concept ;
skos:prefLabel "emotion"@en ;
skos:altLabel "feeling"@en ;
skos:topConceptOf <S> .
```

```
<S> rdf:type skos:ConceptScheme ;
dct:title "My First Thesaurus" ;
skos:hasTopConcept <B> .
```

Цей момент важливий для розуміння формального визначення моделі даних SKOS і як вона може бути реалізована в програмних системах. Цей момент також вкрай важливий для більш розвинених програм Skos, особливо де SKOS і OWL використовуються в поєднанні, як частина гібридноофіційного / напівофіційного дизайну.

З точки зору користувача, різниця між формальною системою подання знань і неофіційною або напівофіційною організацією системи знань, природно, може стати розмитою. Іншими словами, це не може мати відношення до користувача, що <A> і у графі нижче є одиницями (випадки skos:Concept), а <C> і <D> - класами (випадки owl:Class).

```
A> rdf:type skos:Concept ;
skos:prefLabel "love"@en ;
skos:broader <B> .
```

```
<B> rdf:type skos:Concept ;
skos:prefLabel "emotion"@en .
```

```
<C> rdf:type owl:Class ;
rdfs:label "mammals"@en ;
rdfs:subClassOf <D> .
```

```
<D> rdf:type owl:Class ;
rdfs:label "animals"@en .
```

Інформаційна система, яка має будь-яке усвідомлення про інформаційні моделі даних SKOS, тим не менше, повинна враховувати відмінності.

Схеми RDF для SKOS і SKOS eXtension for Labels (SKOS-XL)) наведені в [Додатку С. Документи простору імен SKOS і SKOS-XL](#). Зверніть увагу, що, оскільки існують обмеження, які не можуть бути повністю відображені в схемі, документ RDF / XML містить нормативні підмножини цієї специфікації.

1.4. Сумісність і цілісність

Формальний зміст (*інтерпретація*) графі RDF набуває істинного значення [[RDF-SEMANTICS](#)] [[OWL-SEMANTICS](#)], тобто графа RDF інтерпретується або як істинне, або як помилкове.

Загалом, графа RDF вважається *несумісною*, якщо вона не може бути правдою. Іншими словами, графа RDF є несумісною, якщо вона містить у собі суперечність.

Використання словників RDF і RDFS поодиноці практично унеможлиблює виникнення суперечливої заяви. Коли також використовується словник OWL, існує безліч способів виявити суперечності, наприклад, розглядаючи графу RDF нижче.

```
<Dog> rdf:type owl:Class .  
<Cat> rdf:type owl:Class .  
<Dog> owl:disjointWith <Cat> .  
<dogcat> rdf:type <Dog> , <Cat> .
```

Графа заявляє, що і <Dog>, і <Cat> - класи, і що вони не перетинаються, тобто, що вони не мають спільних членів. Це суперечиться твердженням, що <dogcat> має тип і <Dog>, і <Cat>. Не існує ніякого тлумачення OWL Full, що може задовольнити цю графу, і тому ця графа **не** відповідає OWL Full.

Коли OWL Full використовується в якості мови представлення знань, поняття невідповідності є корисним, оскільки воно розкриває протиріччя в рамках аксіом і фактів, які заявлені в онтології. Завдяки цим невідповідностям ми дізнаємося більше про область знань і приходимо до кращої моделі в цій області, з якої можуть бути зроблені цікаві та значущі висновки.

Коли OWL Full використовується як мова моделювання даних (наприклад, схема), поняття невідповідності знову корисно, але по-іншому. Тут ми не маємо справу з логічною послідовністю людського пізнання себе. Ми просто зацікавлені в офіційному визначенні моделі даних, так що ми можемо встановити з упевненістю чи деякі наведені дані відповідають моделі даних. Якщо дані несумісні з цією моделлю, то дані не підходять.

Тут ми не торкаємося того, чи деякі наведені дані мають будь-яке відношення з реальним світом, тобто чи є вони істинними або помилковими в будь-якому абсолютному значенні. Нас просто цікавить, чи дані відповідають тій чи іншій моделі, так як взаємодія в рамках даного класу може залежати від даних, відповідно до загальної моделі даних.

Ще один спосіб висловити цю думку через поняття *цілісності*. Умови цілісності є формально визначеними моделями даних, які використовуються для ствердження того факту, чи наведені дані узгоджуються з моделлю даних, наприклад, твердження,

що <Dog> і <Cat> - непересічні класи можна розглядати як умову цілісності в моделі даних. При цьому умови, дані нижче, не сумісні.

```
<dogcat> rdf:type <Dog> , <Cat>
```

Визначення моделі даних SKOS, наведених у цьому документі містить обмежену кількість тверджень, які призначені як умови цілісності. Ці умови цілісності включені з метою забезпечення інтероперабельності, визначивши обставини, при яких дані **не відповідають** моделі даних SKOS. Інструменти можуть бути реалізовані, якщо перевірити чи всі (або деякі) з цих умов цілісності відповідають наведеним даним, і тому чи узгоджуються дані з моделлю даних SKOS.

Ці умови цілісності є частиною офіційного визначення класів і властивостей моделі даних SKOS. Тим не менше, вони представлені окремо від інших частин офіційного визначення, оскільки вони служать іншій меті. Умови цілісності служать, перш за все, щоб встановити чи наведені дані узгоджуються з моделлю даних SKOS. Всі інші твердження, що підпадають під визначення моделі даних SKOS служать **тільки** для підтримки логічних висновків. (Див. також наступний підрозділ.)

Умови цілісності визначаються для моделі даних SKOS таким чином, який не залежить від стратегії їх здійснення, наскільки це можливо. Це тому, що існує кілька різних шляхів, за якими може бути здійснена процедура знаходження невідповідності з моделлю даних SKOS. Невідповідності можуть бути знайдені за допомогою міркувань OWL. Крім того, деякі невідповідності можуть бути знайдені за допомогою функції пошуку для конкретної моделі в межах даних, або за допомогою гібридної стратегії .

Умов цілісності для моделі даних SKOS менше, ніж можна було очікувати, особливо для тих, хто звик працювати в замкнутому світі систем управління базами даних. Див також наступний підрозділ.

1.5. Висновки, залежності і відкритий світ припущень

Цей документ визначає модель даних SKOS, як онтологію OWL Full. Є й інші способи, в яких модель даних SKOS можна було б визначити, наприклад, як модель-зв'язок або модель класа UML. Хоча OWL Full в якості мови моделювання даних інтуїтивно видається багато в чому схожим на інші підходи моделювання, існує важлива фундаментальна відмінність.

RDF і OWL Full призначені для систем, в яких дані можуть бути широко поширені (наприклад, в Інтернеті). Оскільки таких систем стає все більше, практично неможливо дізнатися, де знаходяться всі дані в системі. Якщо деякі дані виявляються відсутніми, треба думати, загалом, що дані *можуть* існувати в іншому місці в системі. Це припущення, грубо кажучи, відоме як **відкритий світ припущення** [\[OWL-GUIDE\]](#).

На практиці це означає, що для моделі даних, визначених як онтології OWL Full, деякі визначення можуть мати нелогічний сенс. Ніякі висновки не можуть бути зроблені з відсутніх даних. Це ілюструється, наприклад, визначенням `skos:semanticRelation` в [розділі 8](#) нижче. Властивість `skos:semanticRelation` повинна мати домен і діапазон `skos:Concept`. Ці визначення домену і діапазону дають ліцензії на **висновки**. Розглянемо графу нижче.

```
<A> Skos: semanticRelation <B>.
```

Цей випадок, наведений вище веде за собою наступну графу.

```
<A> rdf:type skos:Concept .  
<B> rdf:type skos:Concept .
```

Таким чином, нам не потрібно *прямо* вказувати, що тут <A> і є прикладами skos:Concept, тому що такі твердження впливають з визначення skos:semanticRelation.

У моделі даних SKOS, більшість тверджень за визначенням **не** є умовами цілісності, а є твердженнями логічних залежностей між різними елементами моделі даних, яка (згідно з відкритим світом припущень) дає дозвіл на ряд простих висновків. Це ілюструється, наприклад, у твердженні в [розділі 7](#) нижче, що skos:broader і skos:narrower є зворотними. Це твердження означає, що

```
<A> skos:narrower <B> .
```

тягне за собою

```
<B> skos:broader <A> .
```

Обидві ці графи, самі по собі, відповідають моделі даних SKOS.

Системи організації знання, такі як тезауруси і класифікаційні схеми застосовуються в самих різних ситуаціях, і окремі системи організації знань можуть бути використані в різних інформаційних системах. Визначивши модель даних SKOS як онтологій OWL Full, Semantic Web може бути використана як засіб для публікації, обміну та зв'язку даних з цими системами організації знань. З цієї причини, за виразністю OWL Full як мови моделювання даних, а також через можливість використання тезаурусів, класифікаційних схем тощо, пліч-о-пліч з офіційною онтологією, OWL Full використовується для визначення моделі даних SKOS . Світ відкритих припущення є основною передумовою моделі даних Skos, і це слід мати на увазі при читанні цього документа.

Див також [\[RDF-PRIMER\]](#) і [\[OWL-GUIDE\]](#).

1.6. Обґрунтування дизайну

Як вже зазначалося вище, поняття системи організації знань охоплює широкий спектр можливостей. Таким чином, існує небезпека надмірності у схемі Skos, що можна було б запобігти застосуванням SKOS в деяких додатках. Для того щоб полегшити це, в ситуаціях, коли виникають сумніви з приводу включення офіційних обмежень (див., наприклад, дискусія про skos:hasTopConcept), обмеження не були заявлені офіційно. У деяких випадках, не дивлячись на те, що ніякого офіційного обмеження немає, рекомендується використання конвенцій. Додатки, що вимагають більш обмеженої поведінки можуть визначити розширення словника SKOS. Див також [\[SKOS-PRIMER\]](#).

1.7. Як читати цей документ

Цей документ формально визначає просту систему організації знань моделі даних як онтології OWL Full. Елементами моделі даних SKOS є класи і властивості OWL, а Uniform Resource Identifier (URI) надаються для кожного з цих класів і властивостей так, щоб їх можна було використати однозначно в мережі Інтернет. Ця сукупність URI є словником SKOS.

Повні словники SKOS наводяться в розділі 2 нижче. Розділи з 3 до 10 формально визначають модель даних SKOS. Визначення моделі даних розбивається на декілька розділів просто для зручності. Кожен з цих розділів від 3 до 10 має наступну загальну схему:

- **Преамбула** - основні ідеї, що описуються в розділі, вводяться в неофіційному порядку.
- **Словник** - URI з словника SKOS, які наведені в розділі.
- **Визначення класів & властивостей** - логічна характеристика і взаємозалежності між класами і властивостями позначених тими URI, які є формально визначеними .
- **Умови цілісності** - якщо є будь-які умови цілісності, вони даються тут.
- **Приклади** – дано деякі канонічні приклади, **які узгоджуються** з моделлю даних Skos, та (за необхідності) приклади, які **не узгоджуються** з моделлю даних SKOS.
- **Нотатки** - будь-які подальші повідомлення і дискусії.

1.7.1. Формальні визначення

Більшість визначень класів і властивостей, і умови цілісності, викладені в даному документі, можуть бути сформульовані як трійки RDF, використовуючи словники RDF, OWL і RDFS. Тим не менше, невелика кількість не може, або з-за обмежень у виразності OWL Full або через відсутність стандартних URI для деяких класів. Для підвищення загальної легкості читання цього документа, а не суміші трійок RDF та інших позначень, формальні визначення і умови цілісності використовують прозу.

Стиль цієї прози, в цілому відповідає стилю, використаному в [\[RDFS\]](#), і повинен бути зрозумілий читачеві зі знанням RDF і OWL.

Так, наприклад, `ex:Person` є екземплярами `owl:Class` означає

```
ex:Person rdf:type owl:Class .
```

`ex:hasParent` і `ex:hasMother` є екземплярами `owl:ObjectProperty` означає:

```
ex:hasParent rdf:type owl:ObjectProperty .
ex:hasMother rdf:type owl:ObjectProperty .
```

`ex:hasMother` є під-властивістю `ex:hasParent` означає

```
ex:hasMother rdfs:subPropertyOf ex:hasParent .
```

" `rdfs:range` з `ex:hasParent` є класом `ex:Person`" означає:

```
ex:hasParent rdfs:range ex:Person .
```

Там, де деякі формальні аспекти моделі даних SKOS не можуть бути вказані як трійки RDF, використовуючи словники RDF, OWL, RDFS, повинно бути зрозуміло читачеві з загальним уявленням про RDF і семантики OWL, як ці аспекти можуть бути переведені на офіційні умови тлумачення словника RDF (наприклад, з [розділу 5](#), "ресурс X не має більше одного значення `skos:prefLabel` в мові тегів" означає, що для будь-якого ресурсу x, ніякі два члени з безлічі { y | <x,y> is in IEXT(I(skos:prefLabel)) } ділять одні й ті ж мови тегів, де I and IEXT є функції, визначені в [\[RDF-SEMANTICS\]](#))..

1.7.2. URI Скорочення

Повні URI, які згадуються в тексті цього документа в моно шрифті, укладені в кутові дужки, наприклад, <http://example.org/ns/example>. Відносні URI наводяться тим же способом, і є відносними до базового URI <http://example.org/ns/>, наприклад, <example> і <http://example.org/ns/example> є ж URI.

URI також наводяться в тексті цього документа в скороченій формі. Скорочені URI, які цитуються в моно шрифті без кутових дужок повинні бути розширені за допомогою таблиці скорочень нижче.

Таблиця 1. Скорочення URI

URI	Абревіатура
http://www.w3.org/2004/02/skos/core#	skos:
http://www.w3.org/1999/02/22-rdf-syntax-ns#	rdf:
http://www.w3.org/2000/01/rdf-schema#	rdfs:
http://www.w3.org/2002/07/owl#	owl:

Так, наприклад, skos:Concept є абревіатурою від <http://www.w3.org/2004/02/skos/core#Concept>

1.7.3. Приклади

Приклади графі RDF наведені за допомогою Terse RDF Triple language (Turtle) [\[TURTLE\]](#). Всі приклади починаються з префікса і базових директив URI:

```
@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
@prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#> .
@prefix owl: <http://www.w3.org/2002/07/owl#> .
@prefix skos: <http://www.w3.org/2004/02/skos/core#> .
@base <http://example.org/ns/> .
```

Таким чином, наведений нижче приклад

Приклад 1

```
<MyConcept> rdf:type skos:Concept .
```

еквівалентний наступному документу Turtle

```
@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
@prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#> .
@prefix owl: <http://www.w3.org/2002/07/owl#> .
@prefix skos: <http://www.w3.org/2004/02/skos/core#> .
@base <http://example.org/ns/> .
```

```
<MyConcept> rdf:type skos:Concept .
```

що еквівалентно наступному документу RDF / XML [\[RDF-XML\]](#)

```
<?xml version="1.0" encoding="UTF-8"?>
<rdf:RDF
  xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:skos="http://www.w3.org/2004/02/skos/core#"
  xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
  xmlns:owl="http://www.w3.org/2002/07/owl#"
  xmlns:example="http://example.org/ns/"
  >
```



```
xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
xmlns:skos="http://www.w3.org/2004/02/skos/core#"
xmlns:owl="http://www.w3.org/2002/07/owl#"
xml:base="http://example.org/ns/">
```

```
<skos:Concept rdf:about="MyConcept"/>
```

```
</rdf:RDF>
```

що еквівалентно наступному документу N-TRIPLES [\[NTRIPLES\]](#)

```
<http://example.org/ns/MyConcept> <http://www.w3.org/1999/02/22-rdf-syntax-ns#type>
<http://www.w3.org/2004/02/skos/core#Concept> .
```

Зверніть увагу на використання в Turtle ";", " і ",", для скорочення кількості трійок з однаковою темою. Деякі приклади також використовують синтаксис Turtle "(...)", представляючи колекції RDF.

1.8. Відповідність

Ця специфікація не визначає офіційне поняття відповідності.

Однак графа RDF буде **несумісною** з моделлю даних SKOS, якщо ця графа і модель даних SKOS (як вона визначена нижче офіційно) у сукупності призведуть до логічних протиріч.

Там, де URI використовується для ідентифікації ресурсів типу `skos:Concept`, `skos:ConceptScheme`, `skos:Collection` або `skosxl:Label` ця специфікація **не** вимагає специфічної поведінки. Тим не менше, настійно рекомендується слідувати керівним принципам, наведеними в [\[COOLURIS\]](#) і [\[RECIPES\]](#).

2. Простір імен і словник SKOS

URI для простору імен SKOS це:

- [#](http://www.w3.org/2004/02/skos/core)

Словник SKOS являє собою набір URI, наведений в лівій колонці у таблиці нижче.

Таблиця 1. Словник SKOS

URI	Визначення
<code>skos:Concept</code>	Розділ 3. Клас <code>skos:Concept</code>
<code>skos:ConceptScheme</code>	Розділ 4. Концептуальні схеми
<code>skos:inScheme</code>	Розділ 4. Концептуальні схеми
<code>skos:hasTopConcept</code>	Розділ 4. Концептуальні схеми
<code>skos:topConceptOf</code>	Розділ 4. Концептуальні схеми
<code>skos:altLabel</code>	Розділ 5. Лексичні позначки

skos:hiddenLabel	Розділ 5. Лексичні позначки
skos:prefLabel	Розділ 5. Лексичні позначки
skos:notation	Розділ 6. Позначення
skos:changeNote	Розділ 7. Властивості документації
skos:definition	Розділ 7. Властивості документації
skos:editorialNote	Розділ 7. Властивості документації
skos:example	Розділ 7. Властивості документації
skos:historyNote	Розділ 7. Властивості документації
skos:note	Розділ 7. Властивості документації
skos:scopeNote	Розділ 7. Властивості документації
skos:broader	Розділ 8. Семантичні відносини
skos:broaderTransitive	Розділ 8. Семантичні відносини
skos:narrower	Розділ 8. Семантичні відносини
skos:narrowerTransitive	Розділ 8. Семантичні відносини
skos:related	Розділ 8. Семантичні відносини
skos:semanticRelation	Розділ 8. Семантичні відносини
skos:Collection	Розділ 9. Концептуальні колекції
skos:OrderedCollection	Розділ 9. Концептуальні колекції
skos:member	Розділ 9. Концептуальні колекції
skos:memberList	Розділ 9. Концептуальні колекції
skos:broadMatch	Розділ 10. Властивості співставлення
skos:closeMatch	Розділ 10. Властивості співставлення
skos:exactMatch	Розділ 10. Властивості співставлення
skos:mappingRelation	Розділ 10. Властивості співставлення
skos:narrowMatch	Розділ 10. Властивості співставлення
skos:relatedMatch	Розділ 10. Властивості співставлення

Всі URI в словнику SKOS будуються шляхом додавання місцевої назви (наприклад, "prefLabel") до URI для простору імен SKOS.

Див. також огляд SKOS в [Додатку В](#), і панель швидкого доступу.

3. Клас skos:Concept

3.1. Преамбула

Клас skos:Concept є класом концептів SKOS.

Концепт SKOS може розглядатися як ідея чи поняття; одиниця думки.

Поняття SKOS концепта корисне при описі концептуальної або інтелектуальної структури системи організації знань, і коли мова йде про конкретні ідеї або значення, створені в рамках KOS.

Зверніть увагу, що, оскільки SKOS покликана стати рушійною силою для представлення напівофіційної KOS, наприклад тезаурусів та класифікаційних схем, певна гнучкість була побудована на формальному визначенні цього класу.

Див [\[SKOS-PRIMER\]](#) для отримання більшої кількості прикладів виявлення та опису концептів SKOS.

3.2. Словник

skos:Concept

3.3. Визначення класів & властивостей

S1	skos:Concept є екземпляром owl:Class.
----	---------------------------------------

3.4. Приклади

Наведена нижче графа говорить, що <MyConcept> є концептом SKOS (наприклад, skos:Concept)

Приклад 2 (відповідає)

```
<MyConcept> rdf:type skos:Concept .
```

3.5. Нотатки

3.5.1. SKOS концепти, класи OWL і властивості OWL

Твердження того, що skos:Concept є екземпляром owl:Class **не** створює ніяких додаткових заяв про формальні відносини між класом з SKOS концепта і OWL класами. Рішення **не** створювати таких заяв було зроблено, щоб надати програмам свободу для вивчення різних шаблонів дизайну для роботи з SKOS в поєднанні з OWL.

У прикладі нижче, <MyConcept> є екземпляром skos:Concept і екземпляром owl:Class

Приклад 3 (відповідає)

```
<MyConcept> rdf:type skos:Concept , owl:Class .
```

Цей приклад **відповідає** моделі даних SKOS.

Аналогічним чином, ця специфікація **не** робить ніяких заяв щодо формальних відносин між класом з SKOS концепта і класом з OWL властивостей.

У прикладі нижче <MyConcept> є екземпляром skos:Concept і екземпляром owl:ObjectProperty

Приклад 4 (відповідає)

```
<MyConcept> rdf:type skos:Concept , owl:ObjectProperty .
```

Цей приклад **відповідає** моделі даних SKOS.

4. Концептуальна схема

4.1. Преамбула

Концептуальна схема SKOS може розглядатися як агрегування одного або декількох SKOS концептів. Семантичні відносини (зв'язки) між цими поняттями можна також розглядати як частину концептуальної схеми. Це визначення не має обмежувального характеру, і існує певна гнучкість у формальній моделі даних, викладеній нижче.

Поняття концептуальної схеми корисне при роботі з даними з невідомих джерел, і коли мова йде про дані, що описують дві або більше різних систем організації знань.

Див в [\[SKOS-PRIMER\]](#) додаткові приклади опису концептуальної схеми.

4.2. Словник

skos:ConceptScheme

skos:inScheme

skos:hasTopConcept

skos:topConceptOf

4.3. Визначення класів & властивостей

S2	skos:ConceptScheme є екземпляром owl:Class.
S3	skos:inScheme, skos:hasTopConcept і skos:topConceptOf є екземплярами owl:ObjectProperty.
S4	rdfs:range з skos:inScheme є класом skos:ConceptScheme.
S5	rdfs:domain з skos:hasTopConcept є класом skos:ConceptScheme.
S6	rdfs:range з skos:hasTopConcept є класом skos:Concept.
S7	skos:topConceptOf є під-властивістю skos:inScheme.
S8	skos:topConceptOf є owl:inverseOf властивістю skos:hasTopConcept.

4.4. Умови цілісності

S9	skos:ConceptScheme не перетинається з skos:Concept.
----	---

4.5. Приклади

Наведена нижче графа описує концептуальну схему з двома поняттями Skos, один з яких є надрівневим концептом в цій схемі.

Приклад 5 (відповідає)

```
<MyScheme> rdf:type skos:ConceptScheme ;
  skos:hasTopConcept <MyConcept> .

<MyConcept> skos:topConceptOf <MyScheme> .

<AnotherConcept> skos:inScheme <MyScheme> .
```

4.6. Нотатки

4.6.1. Закриті системи проти відкритих

Поняття індивідуальної концептуальної схеми SKOS **приблизно відповідає** поняттю індивідуального тезауруса, класифікаційної схеми або інших систем організації знань.

Однак, у більшості діючих інформаційних системах, тезауруси або класифікаційні схеми розглядаються як **закрита система** - розробка концептуальних одиниць, визначених у рамках цієї системи не може брати участь в інших системах (хоча вони можуть *бути* зіставлені з підрозділами в інших системах).

Хоча SKOS дотримується такого підходу, **немає ніяких** умов, що перешкоджають SKOS концепту брати участь в ні одній, одній або більш ніж одній концептуальній схемі.

Так, наприклад, в графі нижче SKOS концепт <MyConcept> бере участь у двох різних концептуальних схемах - **це** узгоджується з моделлю даних SKOS.

Приклад 6 (відповідає)

```
<MyScheme> rdf:type skos:ConceptScheme .

<AnotherScheme> rdf:type skos:ConceptScheme ;
  owl:differentFrom <MyScheme> .

<MyConcept> skos:inScheme <MyScheme> , <AnotherScheme> .
```

Ця гнучкість є бажаною, оскільки вона дозволяє, наприклад, новій концептуальній схемі бути описаною, зв'язуючи дві або більше існуючих концептуальних схем разом.

Відзначимо також, що немає ніякого способу, щоб закрити кордон концептуальної схеми. Таким чином, поки можливо, використовуючи skos:inScheme сказати, що SKOS концепти X, Y і Z беруть участь в концептуальній схемі, немає способу сказати що **тільки** X, Y і Z беруть участь в A.

Тому, хоча SKOS може бути використана **для** опису концептуальної схеми, SKOS не передбачає ніякого механізму, щоб повністю **визначити** концептуальну схему.

4.6.2. Концептуальні схеми SKOS і OWL онтології

Ця специфікація **не** робить ніяких заяв щодо офіційних відносин між класом концептуальних схем SKOS і класом OWL онтологій. Рішення **не** робити такі заяви були зроблені, щоб різні шаблони проектування були вивчені для використання SKOS в поєднанні з OWL [\[OWL-GUIDE\]](#).

У прикладі нижче, <MyScheme> є як концептуальною схемою SKOS так і онтологією OWL. Це узгоджується з моделлю даних SKOS.

Приклад 7 (відповідає)

```
<MyScheme> rdf:type skos:ConceptScheme , owl:Ontology .
```

```
<MyConcept> skos:inScheme <MyScheme> .
```

4.6.3. Головні концепти і смислові відносини

Властивість `skos:hasTopConcept` за умовою, використовується, щоб зв'язати концептуальну схему з SKOS концептом, який є верхнім в ієрархічних відносинах для цієї схеми. Тим не менше, немає ніяких умов дотримання цілісності цієї конвенції. Таким чином, в графі нижче, не будуть суворо дотримуватись використання конвенції для `skos:hasTopConcept`, тим не менше, вона буде **відповідати** моделі даних SKOS.

Приклад 8 (відповідає)

```
<MyScheme> skos:hasTopConcept <MyConcept> .
```

```
<MyConcept> skos:broader <AnotherConcept> .
```

```
<AnotherConcept> skos:inScheme <MyScheme> .
```

Додаток може відкидати такі дані, але це не потрібно.

4.6.4. Схема локалізації та смислові відносини

Зв'язок між двома поняттями SKOS **не** тягне за собою стримування в рамках тієї ж концептуальної схеми. Це показано на прикладі нижче.

Приклад 9 (не відповідає)

```
<A> skos:narrower <B> .
```

```
<A> skos:inScheme <MyScheme> .
```

не тягне за собою

```
<B> skos:inScheme <MyScheme> .
```

Див також [розділ 8](#) нижче.

4.6.5. Домен Skos: inScheme

Зверніть увагу, що **домен не вказує** на властивість `skos:inScheme` тобто домен є класом всіх видів ресурсів (`rdfs:Resource`). Рішення не заявляти про будь-який домен було зроблено, щоб забезпечити певну гнучкість, яка дозволить розширити SKOS, щоб визначати нові класи ресурсів, але все ще використовувати `skos:inScheme`, щоб зв'язати їх з `skos:ConceptScheme` Див також [приклад 82](#) нижче.

5. Лексичні позначки

5.1. Преамбула

Лексичні позначки є рядком символів UNICODE, такі як "романтична любов" або "れんあい", даних природною мовою, наприклад, англійською або японською мовою (тут написано мовою хірагана).

Проста система організації знань дає деякі основні словники для пов'язування лексичних позначок з ресурсами будь-якого типу. Зокрема, SKOS дозволяє розмежування між привілейованими, альтернативними і "прихованими" лексичними позначками для будь-якого ресурсу.

Привілейовані і альтернативні позначки корисні для створення людського розуміння системи організації знань. Ці позначки надають найсильніші підказки про значення поняття SKOS.

Приховані позначки корисні, коли користувач взаємодіє з системою організації знань за допомогою текстового пошуку. Користувач може, наприклад, ввести неправильне написання цього слова, коли намагається знайти відповідний концепт. Якщо неправильне написання запиту може бути зіставлене з прихованою позначкою, користувач зможе знайти відповідний концепт, але приховану позначку не буде видно користувачеві (тому подальші помилки не рекомендуються).

Формально, лексичні позначки є простими літералами RDF [\[RDF-Concepts\]](#). Простий літерал RDF складається з лексичної форми, яка представляє собою рядок символів UNICODE і необов'язкової мови тегів, яка являє собою рядок символів відповідного синтаксису, що визначається як [\[BCP47\]](#).

Див [\[SKOS-PRIMER\]](#) для більшої кількості прикладів позначок SKOS концептів. Слід звернути особливу увагу, що наведені нижче приклади служать тільки для ілюстрації загальних рис моделі даних Skos, і не обов'язково свідчать про найкращу практику для забезпечення позначок з різними мовами тегів. Посилання SKOS спрямоване на створення моделі даних, яка застосовується в цілому ряді ситуацій, які потім можуть бути уточнені і / або обмежені використанням конвенцій для більш конкретних ситуацій.

5.2. Словник

skos:prefLabel

skos:altLabel

skos:hiddenLabel

5.3. Визначення класів & властивостей

S10	skos:prefLabel, skos:altLabel і skos:hiddenLabel є екземплярами owl:AnnotationProperty.
S11	skos:prefLabel, skos:altLabel і skos:hiddenLabel є суб-властивістю rdfs:label.
S12	rdfs:range з кожного з skos:prefLabel, skos:altLabel і skos:hiddenLabel є класом простих літералів RDF

5.4. Умови цілісності

S13	skos:prefLabel, skos:altLabel і skos:hiddenLabel є властивостями, які попарно не перетинаються.
S14	Ресурс має не більше одного значення skos:prefLabel на кожну мову тегів.

5.5. Приклади

Наступна графа **відповідає** SKOS, і свідчить про подання лексичних позначок двома різними мовами (французька та англійська).

Приклад 10 (відповідає)

```
<MyResource>
skos:prefLabel "animals"@en ;
skos:altLabel "fauna"@en ;
skos:hiddenLabel "aminals"@en ;
skos:prefLabel "animaux"@fr ;
skos:altLabel "faune"@fr .
```

Наступна графа **відповідає** SKOS і свідчить про подання лексичних позначок в чотирьох різних варіаціях (написана з японськими ієрогліфами, мовою хірагана, мовою катакана або латинськими літерами (Romaji)).

Приклад 11 (відповідає)

```
<AnotherResource>
skos:prefLabel "東"@ja-Hani ;
skos:prefLabel "ひがし"@ja-Hira ;
skos:altLabel "あずま"@ja-Hira ;
skos:prefLabel "ヒガシ"@ja-Kana ;
skos:altLabel "アスマ"@ja-Kana ;
skos:prefLabel "higashi"@ja-Latn ;
skos:altLabel "azuma"@ja-Latn .
```

Наведена нижче графа **не відповідає** моделі даних Skos, тому що дві різні лексичні позначки були дані тією ж мовою тегів.

Приклад 12 (не відповідає)

```
<Love> skos:prefLabel "love"@en ; skos:prefLabel "adoration"@en .
```

Наведена нижче графа **не відповідає** моделі даних Skos, тому що виникає розбіжність між привілейованими і альтернативними лексичними позначками.

Приклад 13 (не відповідає)

```
<Love> skos:prefLabel "love"@en ; skos:altLabel "love"@en .
```

Наведена нижче графа **не відповідає** моделі даних Skos, тому що виникає розбіжність між альтернативними і прихованими лексичними позначками.

Приклад 14 (не відповідає)

```
<Love> skos:altLabel "love"@en ; skos:hiddenLabel "love"@en .
```

Наведена нижче графа **не відповідає** моделі даних Skos, тому що виникає розбіжність між привілейованими і прихованими лексичними позначками.

Приклад 15 (не відповідає)

```
<Love> skos:prefLabel "love"@en ; skos:hiddenLabel "love"@en .
```

5.6. Нотатки

5.6.1. Домен властивостей лексичних позначок SKOS

Зверніть увагу, що **домен не вказується** для skos:prefLabel, skos:altLabel і skos:hiddenLabel. Таким чином, домен цих властивостей є класом всіх видів ресурсів (rdfs:Resource).

Тому, використовуючи властивості skos:prefLabel, skos:altLabel і skos:hiddenLabel **для** позначення **будь-якого типу ресурсів відповідає** моделі даних SKOS.

У прикладі нижче skos:prefLabel, skos:altLabel і skos:hiddenLabel були використані для позначення ресурсу типу owl:Class - **це** узгоджується з моделлю даних SKOS.

Приклад 16 (відповідає)

```
<MyClass> rdf:type owl:Class ;  
  skos:prefLabel "animals"@en ;  
  skos:altLabel "fauna"@en ;  
  skos:hiddenLabel "aminals"@en ;  
  skos:prefLabel "animaux"@fr ;  
  skos:altLabel "faune"@fr .
```

5.6.2. Діапазон властивостей лексичних позначок SKOS

Зверніть увагу, що діапазон skos:prefLabel, skos:altLabel і skos:hiddenLabel є класом простих літералів RDF. [\[RDF-Concepts\]](#).

За угодою, прості літерали RDF завжди використовуються в об'єкті з позиції трійки, де предикат є одним з skos:prefLabel, skos:altLabel або skos:hiddenLabel. Якщо **граф не** дотримуються використання цієї конвенції, додаток може відкидати такі дані, але це не потрібно. Див також примітку нижче.

5.6.3. Визначення відносин між позначками

Деякі програми вимагають додаткових функцій, пов'язаних з позначками, наприклад, що дозволяють дати опис цих позначок або визначають додаткові зв'язки між позначками (такі як скорочення). Це може бути досягнуто шляхом виявлення позначок за допомогою URI. SKOS eXtension for Labels, визначене в [Додатку А](#), забезпечує підтримку цього.

5.6.4. Альтернативи без привілейованих

В графі нижче, ресурс має дві альтернативні лексичні позначки, але не привілейовані лексичні позначки. **Це** узгоджується з моделлю даних Skos. Однак слід зазначити, що багато додатків потребують привілейованих лексичних позначок з метою отримання оптимального людського розуміння.

Приклад 17 (відповідає)

```
<Love> skos:altLabel "adoration"@en , "desire"@en .
```

5.6.5. Позначки і мови тегів

[BCP47] визначає теги для визначення мови. Зверніть увагу, що "en", "en-GB", "en-US" - три різні мови тегів, що використовуються з англійською мовою, британською англійською та американською англійською, відповідно. Аналогічним чином "ja", "ja-Hani", "ja-Hira", "ja-Kana" та "ja-Latn" - п'ять різних мов тегів, що використовуються з японською, японською з ієрогліфами, мовою хірагана, мовою катакана або латинськими літерами (Romaji) відповідно.

Наведена нижче графа **відповідає** моделі даних Skos, тому що "en", "en-US" and "en-GB" - різні мови тегів.

Приклад 18 (відповідає)

```
<Colour> skos:prefLabel "color"@en , "color"@en-US , "colour"@en-GB .
```

В графі нижче немає жодної розбіжності між лексичними властивостями позначок, знову, тому що "EN" і "EN-GB" різні мови тегів і, отже, графа **відповідає** моделі даних SKOS.

Приклад 19(відповідає)

```
<Love> skos:prefLabel "love"@en ; skos:altLabel "love"@en-GB .
```

Зауважимо однак, що, як зазначено вище в розділі 5.1, ці приклади є тільки ілюстрацією загальних рис моделі даних Skos, і **не** обов'язково свідчать про найкращу практику для забезпечення позначок з різними мовами тегів. .

Пропонується, що додатки потребують позначок в тій чи іншій мові, які відповідають позначкам мови тегів, які передбачені концептуальною схемою SKOS. Програми, які виконуються відповідним чином не потребують, щоб позначки були забезпечені на всіх можливих мовах (яких не може бути багато), і були сумісні з концептуальною схемою SKOS, яка надає тільки ті позначки, лексична форма яких є специфічною для даної мови.

6. Позначення

6.1. Преамбула

Позначення є рядком символів, таких як "T58.5" або "303,4833", який використовується, щоб однозначно визначити концепт в рамках даної концептуальної схеми.

Позначення відрізняється від лексичної позначки в тому, що позначення зазвичай не розпізнаються як слова або послідовності слів будь-якою природною мовою.

Цей розділ визначає властивості skos:notation . Ця властивість використовується для присвоєння позначень [RDF-Concepts].

6.2. Словник

skos:notation

6.3. Визначення класів & властивостей

S15

skos:notation є екземпляром owl:DatatypeProperty.

6.4. Приклади

Приклад нижче ілюструє ресурс `<http://example.com/ns/MyConcept>` з позначеннями, лексичні форми яких є рядком UNICODE "303.4833" і тип даних яких позначається за допомогою URI `<http://example.com/ns/MyNotationDatatype>`

Приклад 20 (відповідає)

```
<MyConcept> skos:notation "303.4833"^^<MyNotationDatatype> .
```

6.5. Нотатки

6.5.1. Позначення, типізовані літерали і типи даних

Типізовані літерали є рядком UNICODE у поєднанні з типом даних URI [\[RDF-Concepts\]](#).

Типізовані літерали зазвичай використовуються для позначення таких чисел, як цілі числа, числа з плаваючою точкою і дати, і існує цілий ряд типів даних, попередньо визначених у специфікації XML Schema [\[XML-SCHEMA\]](#), таких як `xs:integer`, `xs:float` і `xs:date`

В інших ситуаціях, можуть бути визначені нові типи даних , і вони зазвичай називаються "користувацькими типами даних" [\[SWBP-Datatypes\]](#).

Зазвичай, властивість `skos:notation` використовується тільки з типізованими літералами в потрібному положенні, де тип даних URI позначає визначений користувачем тип даних, який відповідає певній системі позначень або класифікаційному коду.

В багатьох ситуаціях може бути достатнім спростити тип даних URI для конкретної системи позначень, а також визначити тип даних в неофіційному документі, який описує, яким чином будуються позначення та / або дозволені лексичні форми. Однак слід зазначити, що також можливо визначити лексичний простір типу даних більш формально за допомогою мови XML Schema, див. [\[SWBP-Datatypes\]](#) розділ 2. Користувачі повинні знати, що інструмент може варіюватися в підтримці типів даних.

6.5.2. Кілька позначень

Немає ніяких обмежень на потужність властивості `skos:notation`. Концепт може мати нуль, 1 або більше умовних позначень.

Де концепт має більш ніж 1 позначення, він може бути з однієї або різних систем числення. У випадку, якщо позначення використовуються з різних систем, різні типи даних можуть використовуватися для їх визначення. Прийнято не призначати більше одного позначення з однієї системи числення (тобто з тим же типом даних URI).

6.5.3. Унікальні позначення в концептуальних схемах

Зазвичай, ніяким двом концептам в тій же концептуальній схемі не можуть бути надані ті ж позначення. Якщо вони будуть дані, то буде неможливо використовувати позначення, що однозначно відносяться до концепта (наприклад, запис стане неоднозначним).

6.5.4. Позначення і привілейовані позначки

Немає ніяких обмежень на спільне використання `skos:notation` і `skos:prefLabel`. У прикладі нижче, тим же рядком задається як лексична форма позначення, так і лексична форма привілейованих позначок.

Приклад 21 (відповідає)

```
<Potassium>  
skos:prefLabel "K"@en ;  
skos:notation "K"^^<ChemicalSymbolNotation> .
```

Типізовані літерали складаються з рядка символів і типу даних URI. Зазвичай, `skos:notation` використовується тільки з **типізованими літералами** в позиції трійки.

Прості літерали складаються з рядка символів та мовних тегів. Зазвичай, `skos:prefLabel` (і `skos:altLabel`, і `skos:hiddenLabel`) застосовуються тільки з **простими літералами** в позиції трійки.

Не існує такого поняття, як RDF літерал з мовними тегами і з типом даних URI, наприклад, типізовані літерали не мають мовних тегів, а текстові літерали не мають типу даних URI.

6.5.5. Сфера `skos:notation`

Зверніть увагу, що ні один **домен не вказується** для `skos:notation`. Таким чином, домен є класом всіх видів ресурсів (`rdfs:Resource`). Тому, використання `skos:notation` з будь-яким типом ресурсів відповідає моделі даних SKOS.

7. Властивості документації (Властивості нотатки)

7.1. Преамбула

Нотатки використовуються для надання інформації, що стосується SKOS концептів. Не існує ніяких обмежень щодо характеру цієї інформації, наприклад, це може бути просто текст, гіпертекст, або просто картинка, це може бути визначення, інформація про масштаби концептів, редакційна інформація, або будь-які інші види інформації.

Є сім властивостей в SKOS для пов'язування нотатки з концептом, які визначаються в цьому розділі. Додаткову інформацію про використання кожної з властивостей документації SKOS див. [\[SKOS-PRIMER\]](#).

Ці сім властивостей не мають на меті охопити всі ситуації, а, швидше, бути корисними в деяких з найбільш поширених ситуацій, і забезпечити систему точок розширення для визначення більш конкретних типів нотатки. Більш докладна інформація про розширення Skos див. [\[SKOS-PRIMER\]](#).

Три різні моделі використання рекомендуються в [\[SKOS-PRIMER\]](#) для властивостей документації SKOS - "документація, як RDF літерал", "документація, як опис ресурсу" і "документація, як довідковий документ". Моделі даних, визначені в цьому розділі, призначені для розміщення всіх трьох моделей використання.

7.2. Словник

skos:note

skos:changeNote

skos:definition

skos:editorialNote

skos:example

skos:historyNote

skos:scopeNote

7.3. Визначення класів & властивостей

S16	skos:note, skos:changeNote, skos:definition, skos:editorialNote, skos:example, skos:historyNote і skos:scopeNote є екземплярами owl:AnnotationProperty.
S17	skos:changeNote, skos:definition, skos:editorialNote, skos:example, skos:historyNote і skos:scopeNote є над-властивістю skos:note.

7.4. Приклади

Наведена нижче графа дає приклад "документації, як RDF літерала".

Приклад 22 (відповідає)

```
<MyResource> skos:note "this is a note"@en .
```

Наведена нижче графа дає приклад "документації, як довідкового документа".

Приклад 23 (відповідає)

```
<MyResource> skos:note <MyNote> .
```

7.5. Нотатки

7.5.1. Сфера властивостей документації SKOS

Зверніть увагу, що ні один **домен не вказується** для властивостей документації SKOS. Таким чином, домен є класом всіх видів ресурсів (rdfs:Resource). Тому, використання властивостей документації SKOS для надання інформації з **будь-яких видів ресурсів** відповідає моделі даних SKOS.

У прикладі нижче, skos:definition використовується для забезпечення простого текстового визначення ресурсу типу owl:Class - це узгоджується з моделлю даних SKOS.

Приклад 24 (відповідає)

```
<Protein> rdf:type owl:Class ;  
  skos:definition ""A physical entity consisting of a sequence of amino-acids; a protein monomer; a single polypeptide chain. An example is the EGFR protein.""@en .
```

7.5.2. Діапазон властивостей документації SKOS

Зауважимо, що ні один діапазон не вказує про властивості документації SKOS, і, отже, діапазон цих властивостей є ефективним класом всіх видів ресурсів (rdfs:Resource). У RDF і OWL Full semantics все є ресурсом, включаючи прості літерали RDF.

8. Семантичні відносини

8.1. Преамбула

SKOS семантичні відносини пов'язані з SKOS концептами.

Проста система організації знань розрізняє дві основні категорії семантичних відносин: **ієрархічні** та **асоціативні**. Ієрархічні зв'язки між двома концептами показують, що один є більш загальним ("Розширений"), ніж інші ("вужькі"). Асоціативний зв'язок між двома концептами вказує, що обидва є "пов'язані", але один ні в якій мірі не більш загальний, ніж інший.

Властивості skos:broader і skos:narrower використовуються для ствердження прямих ієрархічних зв'язків між двома SKOS концептами. Трійка <A> skos:broader стверджує, що , об'єкт трійки, є більш широке поняття, ніж <A>, суб'єкт трійки. Аналогічним чином, трійка <C> skos:narrower <D> стверджує, що <D>, об'єкт трійки, є більш вужьким поняттям, ніж <C>, суб'єкт трійки.

Зазвичай, skos:broader і skos:narrower використовуються **тільки** для ствердження **прямих** (тобто безпосередніх) ієрархічних зв'язків між двома SKOS концептами. Це забезпечує застосування програм зі зручним і надійним способом доступу до прямих ширших і вужьких зв'язків будь-якого концепта. Зауважимо, що для підтримки використання цієї конвенції, властивості skos:broader і skos:narrower **не оголошені** транзитивними властивостями.

В деяких програмах необхідно використовувати **як прямі, так і непрямі** ієрархічні зв'язки між концептами, наприклад, для покращення пошуку шляхом розширення запиту. З цією метою, забезпечуються властивості skos:broaderTransitive і skos:narrowerTransitive. Трійка <A> skos:broaderTransitive являє собою прямі або

опосередковані ієрархічні зв'язки, де є більш ширшим "предком" <A>. Аналогічно трійка <C> skos:narrowerTransitive <D> являє собою прямі або опосередковані ієрархічні зв'язки, де <D> є вузким "нащадком" <C>.

Зазвичай, властивостям skos:broaderTransitive і skos:narrowerTransitive **не** потрібно робити затверджень. Скоріше, ці властивості використовуються для виведення транзитивного закриття ієрархічних зв'язків, які потім можуть бути використані для доступу до прямих або непрямих ієрархічних зв'язків між концептами.

Властивість skos:related використовується, щоб стверджувати асоціативний зв'язок між двома концептами SKOS.

Для більшої кількості прикладів ієрархічних та асоціативних зв'язків див. [\[SKOS-PRIMER\]](#).

8.2. Словник

skos:semanticRelation

skos:broader

skos:narrower

skos:related

skos:broaderTransitive

skos:narrowerTransitive

8.3 Визначення класів & властивостей

S18	skos:semanticRelation, skos:broader, skos:narrower, skos:related, skos:broaderTransitive і skos:narrowerTransitive є екземпляром owl:ObjectProperty.
S19	rdfs:domain з skos:semanticRelation є класом skos:Concept.
S20	rdfs:range з skos:semanticRelation є класом skos:Concept.
S21	skos:broaderTransitive, skos:narrowerTransitive і skos:related є під-властивостями skos:semanticRelation.
S22	skos:broader є суб-властивостями skos:broaderTransitive, і skos:narrower є під-властивостями skos:narrowerTransitive.
S23	skos:related є екземпляром owl:SymmetricProperty.
S24	skos:broaderTransitive і skos:narrowerTransitive є екземпляром owl:TransitiveProperty.
S25	skos:narrower є owl:inverseOf властивості skos:broader.
S26	skos:narrowerTransitive є owl:inverseOf властивості skos:broaderTransitive.

8.4. Умови цілісності

Зверніть увагу, що skos:related є симетричною властивістю, і skos:broaderTransitive і skos:narrowerTransitive є зворотними, skos:related також не перетинається з skos:narrowerTransitive .

8.5. Приклади

Наведена нижче графа стверджує ієрархічний прямиий зв'язок між <A> і (де ширше, ніж <A>) і асоціативний зв'язок між <A> і <C>, і **відповідає** моделі даних SKOS .

Приклад 25 (відповідає)

```
<A> skos:broader <B> ; skos:related <C> .
```

Наведений нижче приклад **не узгоджується** з моделлю даних Skos, тому що є розбіжність між асоціативними зв'язками та ієрархічними зв'язками.

Приклад 26 (не відповідає)

```
<A> skos:broader <B> ; skos:related <B> .
```

Наведений нижче приклад **не узгоджується** з моделлю даних Skos, тому що знову є розбіжність між асоціативними зв'язками та ієрархічними зв'язками.

Приклад 27 (не відповідає)

```
<A> skos:broader <B> ; skos:related <C> .  
<B> skos:broader <C> .
```

У наведеному вище прикладі, розбіжність не відразу очевидна. Розбіжність стає очевидною, коли зроблені висновки, засновані на визначенні класів і властивостей вище.

Приклад 28 (не відповідає)

```
<A> skos:broaderTransitive <C> ; skos:related <C> .
```

Наведений нижче приклад **не узгоджується** з моделлю даних Skos, тому що знову є розбіжність між асоціативними зв'язками та ієрархічними зв'язками, яка може впливати з визначення класів і властивостей, наведених вище.

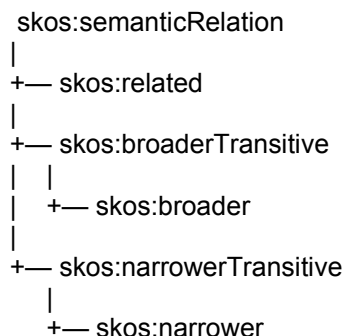
Приклад 29 (не відповідає)

```
<A> skos:narrower <B> ; skos:related <C> .  
<B> skos:narrower <C> .
```

8.6. Нотатки

8.6.1. Відносини під-властивостей

Наведена нижче діаграма ілюструє неофіційні відносини під-властивостей між семантичними властивостями відносин SKOS.



8.6.2. Домен і діапазон властивостей семантичних відносин SKOS

Зауважимо, що домен і діапазон `skos:semanticRelation` є класом `skos:Concept`. Оскільки `skos:broader`, `skos:narrower` і `skos:related` є під-властивістю `skos:semanticRelation`, графа в [прикладі 26](#) вище вказує, що `<A> ` і `<C>` є екземплярами `skos:Concept`

8.6.3 Симетрія `skos:related`

`skos:related` є симетричною властивістю. Приклад нижче ілюструє втілення, яке впливає з цієї умови.

Приклад 30 (втілення)

```
<A> skos:related <B> .
втільює
<B> skos:related <A> .
```

Зауважимо, що хоча `skos:related` є симетричною властивістю, ця умова **не накладає** жодних обмежень на під-властивості `skos:related` (тобто під-властивість `skos:related` може бути симетрична, несиметрична або антисиметрична, і відповідати моделі даних SKOS).

Щоб проілюструвати цей момент, у наведеному нижче прикладі, дві нові властивості, які **не** є симетричними, оголошені як під-властивості `skos:related`. Приклад, **що** узгоджується з моделлю даних Skos, також показує деякі втілення.

Приклад 31 (втілення)

```
<cause> rdf:type owl:ObjectProperty ;
  rdfs:subPropertyOf skos:related .

<effect> rdf:type owl:ObjectProperty ;
  rdfs:subPropertyOf skos:related ;
  owl:inverseOf <cause> .

<A> <cause> <B> .
втільює
<A> skos:related <B> .

<B> <effect> <A> ; skos:related <A> .
```

Див також [\[SKOS-PRIMER\]](#) для огляду рекомендацій з розширення SKOS.

8.6.4. skos:related і транзитивність

Зверніть увагу, що skos:related **не** є транзитивною властивістю. Таким чином, модель даних SKOS **не підтримує** втілення, як показано в наступному прикладі.

Приклад 32 (без втілення)

```
<A> skos:related <B> .  
<B> skos:related <C> .  
не втілює  
<A> skos:related <C> .
```

8.6.5. skos:related і рефлексивність

Зауважимо, що ця специфікація не стверджує, що skos:related є рефлексивною властивістю, але також не стверджує, що skos:related є іррефлексивною властивістю.

Оскільки skos:related **не** визначений як іррефлексивна властивість, приклад нижче **узгоджується** з моделлю даних SKOS.

Приклад 33 (відповідає)

```
<A> skos:related <A> .
```

Однак, для багатьох додатків, які використовують системи організації знань, твердження в формі $X \text{ skos:related } X$ є потенційною проблемою. Якщо це так, додаток може потребувати пошуку таких тверджень для обробки даних SKOS.

8.6.6. skos:broader і транзитивність

Зверніть увагу, що skos:broader **не** є транзитивною властивістю. Аналогічним чином, skos:narrower **не** є транзитивною властивістю. Таким чином, модель даних SKOS **не підтверджує** втілення як показано в наступному прикладі.

Приклад 34 (без втілення)

```
<A> skos:broader <B> .  
<B> skos:broader <C> .  
не втілює  
<A> skos:broader <C> .
```

Однак, skos:broader є під-властивістю skos:broaderTransitive, яка є транзитивною властивістю. Аналогічним чином, skos:narrower є під-властивістю skos:narrowerTransitive, яка є транзитивною властивістю. Таким чином, модель даних SKOS підтримує втілення, яке проілюстровано нижче.

Приклад 35 (втілення)

```
<A> skos:broader <B> .  
<B> skos:broader <C> .  
втілює  
<A> skos:broaderTransitive <B> .  
<B> skos:broaderTransitive <C> .  
<A> skos:broaderTransitive <C> .
```


Слід звернути особливу увагу, що, зазвичай, `skos:broader` і `skos:narrower` використовуються **тільки** для негайного ствердження (тобто, прямих) ієрархічних зв'язків між двома SKOS концептами. Зазвичай, для `skos:broaderTransitive` і `skos:narrowerTransitive` **не потрібно** робити затверджень, а замість цього використовують висновки.

Ця модель дозволяє зберегти інформацію про прямі (наприклад, негайні) ієрархічні зв'язки, які необхідні для вирішення багатьох задач (наприклад, будуючи різні типи візуального представлення системи організації знань), також забезпечення механізму для зручних запитів транзитивного закриття цих ієрархічних зв'язків (який буде включати в себе прямі і непрямі зв'язки), які корисні і в інших ситуаціях (наприклад, розширення запиту алгоритму).

Відзначимо також, що під-властивість транзитивної властивості **не** обов'язково транзитивна.

Див також відомості про альтернативні шляхи нижче.

8.6.7. `skos:broader` і рефлексивність

Зауважимо, що ця специфікація не стверджує рефлексивних характеристик `skos:broader`. Не затверджено, що `skos:broader` є рефлексивною властивістю, **але також не затверджено, що** `skos:broader` є іррефлексивною властивістю. Таким чином, для будь-якого прикладу і ресурсу `<A>`, трійка:

Приклад 36 (відповідає)

```
<A> skos:broader <A> .
```

може чи не може бути присутня. Ця консервативна позиція дозволяє SKOS бути використаною для моделювання KOS, де тлумачення `skos:broader` є рефлексивним (наприклад, прямий переклад висновку ієрархії під-класів OWL), або де KOS `skos:broader` можна вважати іррефлексивним (є доцільним для більшості тезаурусів та класифікаційних схем).

Аналогічним чином, немає тверджень щодо рефлексивності або іррефлексивності `skos:narrower`

Однак для багатьох додатків, які використовують системи організації знань, твердження у формі `X skos:broader X` або `Y skos:narrower Y` являють собою потенційну проблему. Якщо це так, додаток може потребувати пошуку таких тверджень для обробки даних SKOS, хоча, як додаток повинен обробляти такі заяви не визначено в цій специфікації, і може варіюватися між додатками.

8.6.8. Цикли в ієрархічних відносинах (`skos:broaderTransitive` і рефлексивність)

В графі нижче, цикл зазначається в ієрархічних відносинах. Зверніть увагу, що ця графа **відповідає** моделі даних SKOS, тобто **немає положення**, що вимагає, щоб `skos:broaderTransitive` було іррефлексивним.

Приклад 37 (відповідає)

```
<A> skos:broader <B> .  
<B> skos:broader <A> .
```

Однак для багатьох застосувань, де використовуються системи організації знань, цикл в ієрархічних відносинах являє собою потенційну проблему. Для таких додатків, обчислюється транзитивне закриття `skos:broaderTransitive`, потім шукається твердження виду `X skos:broaderTransitive X`, що є зручним для знаходження циклів в ієрархічних відносинах. Як додаток повинен обробляти такі твердження не визначено в цій специфікації, і може відхилятися від додатків.

8.6.9. Альтернативні шляхи в ієрархічних відносинах

В графі нижче, існує два альтернативних шляхи від А до С в ієрархічних відносинах.

Приклад 38 (відповідає)

```
<A> skos:broader <B> , <C> .  
<B> skos:broader <C> .
```

В графі нижче, існує два альтернативних шляхи від А до D в ієрархічних відносинах.

Приклад 39 (відповідає)

```
<A> skos:broader <B> , <C> .  
<B> skos:broader <D> .  
<C> skos:broader <D> .
```

Це є зразок, що природно виникає в полі-ієрархічних системах організації знань.

Обидва ці приклади **відповідають** моделі даних SKOS, тобто **немає положення**, що вимагає існування тільки одного шляху між будь-якими двома вузлами в ієрархічних відносинах.

8.6.10. Диз'юнктність `skos:related` та `skos:broaderTransitive`

Ця специфікація розглядає ієрархічні і асоціативні відносини, як принципово різні за своїм характером. Тому розбіжність між ієрархічними та асоціативними зв'язками **не узгоджується** з моделлю даних SKOS. Наведені вище приклади ілюструють деякі ситуації, в яких можуть виникнути розбіжності.

Ця позиція відповідає звичайним визначенням, що містяться в ієрархічних і асоціативних відносинах у стандартах тезауруса [\[ISO2788\]](#) [\[BS8723-2\]](#), а також підтримується поширеною практикою в багатьох існуючих системах організації знань.

Зауважимо, що ця специфікація займає сильні позиції, що не тільки безпосередньо розділяє (наприклад, прямий) ієрархічні та асоціативні зв'язки, але також розділяє асоціативні зв'язки з *непрямими* ієрархічними зв'язками. Ця ідея відображена формально в стані цілісності, стверджуючи, що `skos:related` і `skos:broaderTransitive` є непересічними властивостями.

9. Концептуальні колекції

9.1. Преамбула

Концептуальні колекції SKOS є помічені і / або впорядковані групи концептів SKOS.

Колекції корисні, коли в групі концептів є щось спільне, і є зручним групувати їх під загальною назвою, або коли деякі концепти можуть бути розміщені в конструктивному порядку.

9.2. Словник

skos:Collection

skos:OrderedCollection

skos:member

skos:memberList

9.3. Визначення класів & властивостей

S28	skos:Collection і skos:OrderedCollection є екземпляром owl:Class.
S29	skos:OrderedCollection є під-класом skos:Collection.
S30	skos:member і skos:memberList є прикладом owl:ObjectProperty.
S31	rdfs:domain з skos:member є класом skos:Collection.
S32	rdfs:range з skos:member є об'єднанням класів skos:Concept і skos:Collection.
S33	rdfs:domain з skos:memberList є класом skos:OrderedCollection.
S34	rdfs:range з skos:memberList є класом rdf:List.
S35	skos:memberList є екземпляром owl:FunctionalProperty.
S36	Для будь-якого ресурсу, кожен пункт у наведеному переліку в якості значення skos:memberList властивості є також skos:member властивість.

9.4. Умови цілісності

S37	skos:Collection не розділяється skos:Concept і skos:ConceptScheme.
-----	--

9.5. Приклади

Наведена нижче графа ілюструє колекції SKOS з 3 членами.

Приклад 40 (відповідає)

```
<MyCollection> rdf:type skos:Collection ;  
skos:member <X> , <Y> , <Z> .
```

Наведена нижче графа ілюструє колекції SKOS з 3 членами. Зверніть увагу на використання Turtle синтаксису (...), що представляє колекцію RDF (список).

Приклад 41 (відповідає)

```
<MyOrderedCollection> rdf:type skos:OrderedCollection ;
```

```
skos:memberList ( <X> <Y> <Z> ) .
```

9.6. Нотатки

9.6.1. Визначені колекції з упорядкованих колекцій

Твердження [S36](#) затверджує логічний взаємозв'язок між властивостями `skos:memberList` та `skos:member`. Цей зв'язок означає, що колекція може бути визначена з упорядкованих колекції. Це показано на прикладі нижче.

Приклад 42 (втілення)

```
<MyOrderedCollection> rdf:type skos:OrderedCollection ;  
skos:memberList ( <X> <Y> <Z> ) .
```

втілює

```
<MyOrderedCollection> rdf:type skos:Collection ;  
skos:member <X> , <Y> , <Z> .
```

Зверніть увагу, що SKOS не передбачає будь-який спосіб прямо заявити, що колекція **не** упорядкована.

9.6.2. Цілісність `skos:memberList`

Зверніть увагу, що `skos:memberList` є функціональною властивістю, тобто вона не може мати більше ніж одне значення. Це зроблено для того, щоб захопити в рамках моделі даних SKOS впорядковану колекцію, яка має більш ніж один член. На жаль, немає ніякої можливості використовувати цю умову як умову цілісності. Іншими словами, хоча приклад в графі нижче **відповідає** моделі даних Skos, це є нонсенс (список з двома першими елементами і роздвоєним кінцем).

Приклад 43 (втілення)

```
<OrderedCollectionResource>  
skos:memberList ( <A> <B> ) , ( <X> <Y> ) .
```

втілює

```
<OrderedCollectionResource>  
skos:memberList [ rdf:first <A> , <X> ; rdf:rest [ rdf:first <B> ; rdf:rest rdf:nil ] , [ rdf:first  
<Y> ; rdf:rest rdf:nil ] ] .
```

Однак, як зазначається в [\[RDF-SEMANTICS\]](#) розділі 3.3.3, семантичні розширення RDF можуть розміщувати додатково синтаксичні формальні обмеження на використання словника RDF колекції (`rdf:first rdf:rest rdf:nil`) з метою виключення такої графі.

9.6.3. Вкладені колекції

У прикладі, наведеному нижче, колекція є вкладена в іншу колекцію.

Приклад 44 (відповідає)

```
<MyCollection> rdf:type skos:Collection ;  
skos:member <A> , <B> , <MyNestedCollection> .
```

```
<MyNestedCollection> rdf:type skos:Collection ;  
skos:member <X> , <Y> , <Z> .
```

Цей приклад **відповідає** моделі даних Skos, оскільки діапазон skos:member є об'єднанням skos:Concept і skos:Collection

9.6.4. SKOS концепти, концептуальні колекції і семантичні відносини

У моделі даних Skos, skos:Concept і skos:Collection є непересічні класи. Домен і діапазон властивостей семантичних відносин SKOS є skos:Concept. Тому, якщо будь-які властивості семантичних відносин SKOS (наприклад, skos:narrower) використовуються для посилання або з колекції, графа **не** буде узгоджуватися з моделлю даних SKOS.

Це показано в наступному прикладі, який **не узгоджується** з моделлю даних SKOS.

Приклад 45 (не відповідає)

```
<A> skos:narrower <B> .  
<B> rdf:type skos:Collection .
```

Крім того, приклад нижче, **не узгоджується** з моделлю даних SKOS.

Приклад 46 (не відповідає)

```
<A> skos:broader <B> .  
<B> rdf:type skos:Collection .
```

Крім того, приклад нижче, **не узгоджується** з моделлю даних SKOS.

Приклад 47 (не відповідає)

```
<A> skos:related <B> .  
<B> rdf:type skos:Collection .
```

Однак, графа нижче узгоджується з моделлю даних SKOS.

Приклад 48 (відповідає)

```
<A> skos:narrower <B> , <C> , <D> .  
  
<ResourceCollection> rdfs:label "Resource Collection"@en ;  
skos:member <B> , <C> , <D> .
```

Це означає, що для тезаурусів та інших систем організації знань, де вузлові мітки використовуються в рамках систематичного відображення для цього тезауруса, відповідне подання SKOS вимагає ретельного розгляду. Крім того, де вузлові мітки використовуються в систематичному дисплеї, не завжди може бути можливим повною мірою відновити систематичне відображення SKOS. Повне представлення усіх відомостей, представлених в систематичному відображенні тезауруса або інших систем організації знань, включаючи детальну інформацію про розташування і презентації, виходить за рамки SKOS.

10. Властивості співставлення

10.1. Преамбула

Властивостями співставлення SKOS є skos:closeMatch, skos:exactMatch, skos:broadMatch, skos:narrowMatch і skos:relatedMatch. Ці властивості використовуються для відображення (вирівнювання) зв'язків співставлення між концептами SKOS в різних концептуальних схемах, де зв'язки є невід'ємною частиною змісту.

Властивості skos:broadMatch і skos:narrowMatch використовуються для позначення ієрархічних зв'язків співставлення між двома концептами.

Властивості skos:relatedMatch використовується для позначення асоціативних зв'язків співставлення між двома концептами.

Властивості skos:closeMatch використовується, щоб зв'язати два концепти, які досить близькі, так, щоб вони могли бути використані як взаємозамінні в **деяких** додатках інформаційного пошуку. Для того щоб уникнути можливості "з'єднання помилок" при об'єднанні співставлення більш ніж у двох концептуальних схемах, skos:closeMatch **не** оголошена транзитивною властивістю.

Властивість skos:exactMatch використовується, щоб зв'язати два поняття, що вказують на високу ступінь впевненості, що поняття можуть бути взаємозамінними в широкому спектрі додатків інформаційного пошуку. skos:exactMatch є транзитивною властивістю, і є під-властивістю skos:closeMatch

10.2. Словник

skos:mappingRelation

skos:closeMatch

skos:exactMatch

skos:broadMatch

skos:narrowMatch

skos:relatedMatch

10.3. Визначення класів & властивостей

S38	skos:mappingRelation, skos:closeMatch, skos:exactMatch, skos:broadMatch, skos:narrowMatch and skos:relatedMatch є екземплярами owl:ObjectProperty.
S39	skos:mappingRelation є під-властивістю skos:semanticRelation.
S40	skos:closeMatch, skos:broadMatch, skos:narrowMatch і skos:relatedMatch є під-властивістю skos:mappingRelation.
S41	skos:broadMatch є під-властивістю skos:broader, skos:narrowMatch є під-властивістю skos:narrower, і skos:relatedMatch є під-властивістю skos:related.
S42	skos:exactMatch є під-властивістю skos:closeMatch.
S43	skos:narrowMatch є owl:inverseOf властивістю skos:broadMatch.
S44	skos:relatedMatch, skos:closeMatch і skos:exactMatch є прикладами owl:SymmetricProperty.

S45 skos:exactMatch є прикладами owl:TransitiveProperty.

10.4. Умови цілісності

S46 skos:exactMatch не перетинається з кожним із властивостей skos:broadMatch and skos:relatedMatch.

Зверніть увагу, що skos:exactMatch є симетричною властивістю, skos:broadMatch і skos:narrowMatch є зворотними, skos:exactMatch тому також не перетинаються з skos:narrowMatch.

10.5. Приклади

Наведений нижче приклад стверджує точний зв'язок співставлення еквівалентності між <A> і

Приклад 49 (віدповідає)

<A> skos:exactMatch .

Наведений нижче приклад стверджує тісний зв'язок співставлення еквівалентності між <A> і

Приклад 50 (відповідає)

<A> skos:closeMatch .

Наведений нижче приклад стверджує ієрархічні зв'язки співставлення між <A> і (де є ширшим ніж <A>), і асоціативні зв'язки співставлення між <A> і <C>

Приклад 51 (відповідає)

<A> skos:broadMatch ; skos:relatedMatch <C> .

Наведений нижче приклад **не узгоджується** з моделлю даних Skos, оскільки існує розбіжність між точними і ієрархічними зв'язками співставлення.

Приклад 52 (не відповідає)

<A> skos:exactMatch ; skos:broadMatch .

Наведений нижче приклад **не узгоджується** з моделлю даних Skos, тому що є розбіжність між точними та асоціативними зв'язками співставлення.

Приклад 53 (не відповідає)

<A> skos:exactMatch ; skos:relatedMatch .

10.6. Нотатки

10.6.1. Властивості співставлення, властивості семантичних відносин і концептуальні схеми

Зазвичай, властивості співставлення SKOS використовується тільки для зв'язку концептів в різних концептуальних схемах. Однак слід зазначити, що використання семантичних властивостей SKOS (`skos:broader`, `skos:narrower`, `skos:related`) для зв'язку концептів в різних концептуальних схемах **також** узгоджується з моделлю даних SKOS (див. [розділ 8](#)).

Властивості співставлення `skos:broadMatch`, `skos:narrowMatch` і `skos:relatedMatch` надаються для зручності, для ситуацій, коли відомо про походження даних, і корисно мати можливість відразу ж встановити різницю між внутрішніми зв'язками в рамках концептуальної схеми і зв'язками співставлення між концептуальними схемами.

Однак, використання властивості співставлення SKOS **не замінює** ретельного управління графи RDF або використання механізмів походження.

Сенс цієї конструкції в тому, що важко провести різницю між внутрішніми зв'язками в рамках концептуальної схеми і зв'язками співставлення між концептуальними схемами. Це особливо актуально у відкритому середовищі, де різні люди можуть реорганізувати концепти в концептуальні схеми по-різному. Те, що одна особа переглядає дві концептуальних схеми з зв'язками співставлення між ними, інша може відкрити в одній концептуальній схемі тільки з внутрішніми зв'язками. Ця специфікація дозволяє, щоб обидві точки зору співіснували, що (як можна сподіватися) сприятиме гнучкості та інноваціям в повторному використанні даних SKOS в мережі Інтернет.

Тому існує тісний зв'язок між семантичними відносинами властивостей SKOS і властивостей співставлення SKOS. Властивість `skos:broadMatch` є під-властивість `skos:broader`, `skos:narrowMatch` є під-властивість `skos:narrower`, і `skos:relatedMatch` є під-властивість `skos:related`. Повний набір відносин під-властивостей проілюстровано нижче.

```
skos:semanticRelation
|
+- skos:related
| |
| +- skos:relatedMatch
|
+- skos:broaderTransitive
| |
| +- skos:broader
|   |
|   +- skos:broadMatch
|
+- skos:narrowerTransitive
| |
| +- skos:narrower
|   |
|   +- skos:narrowMatch
|
+- skos:mappingRelation
|
+- skos:closeMatch
| |
| +- skos:exactMatch
|
+- skos:relatedMatch
|
```

+ skos:broadMatch
|
+ skos:narrowMatch

Наведені нижче приклади ілюструють деякі втілення, які впливають з цього дерева під-властивостей, і з області та діапазону skos:semanticRelation

Приклади 54 (втілення)

```
<A> skos:broadMatch <B> .  
втілює  
<A> skos:mappingRelation <B> .  
<A> skos:broader <B> .  
<A> skos:broaderTransitive <B> .  
<A> skos:semanticRelation <B> .  
<A> rdf:type skos:Concept .  
<B> rdf:type skos:Concept .
```

Приклад 55 (втілення)

```
<A> skos:narrowMatch <B> .  
втілює  
<A> skos:mappingRelation <B> .  
<A> skos:narrower <B> .  
<A> skos:narrowerTransitive <B> .  
<A> skos:semanticRelation <B> .  
<A> rdf:type skos:Concept .  
<B> rdf:type skos:Concept .
```

Приклад 56 (втілення)

```
<A> skos:relatedMatch <B> .  
втілює  
<A> skos:mappingRelation <B> .  
<A> skos:related <B> .  
<A> skos:semanticRelation <B> .  
<A> rdf:type skos:Concept .  
<B> rdf:type skos:Concept .
```

Приклад 57 (втілення)

```
<A> skos:exactMatch <B> .  
втілює  
<A> skos:closeMatch <B> .  
<A> skos:mappingRelation <B> .  
<A> skos:semanticRelation <B> .  
<A> rdf:type skos:Concept .  
<B> rdf:type skos:Concept .
```

Відзначимо також, що, оскільки різні люди можуть реорганізувати концепти в концептуальні схеми по-різному, графа може стверджувати зв'язки співставлення між концептами в тій же концептуальній схемі, і **не** існує жодних формальних умов цілісності в моделі даних SKOS, які дозволять зробити цю графу несумісною, наприклад, графа нижче **відповідає** моделі даних SKOS. Однак, на практиці очікується, що така графа виникне в результаті злиття двох або більше граф з різних джерел.

Приклад 58 (відповідає)

```
<A> skos:broadMatch <B> ; skos:relatedMatch <C> .  
  
<A> skos:inScheme <MyScheme> .
```

```
<B> skos:inScheme <MyScheme> .  
<C> skos:inScheme <MyScheme> .
```

10.6.2. Розбіжність між ієрархічними і асоціативними зв'язками

Нижче наводяться приклади "розбіжності" між ієрархічними і асоціативними зв'язками співставлень, які **не узгоджуються** з моделлю даних SKOS .

Приклад 59 (не відповідає)

```
<A> skos:broadMatch <B> ; skos:relatedMatch <B> .
```

Приклад 60 (не відповідає)

```
<A> skos:narrowMatch <B> ; skos:relatedMatch <B> .
```

Приклад 61 (не відповідає)

```
<A> skos:broadMatch <B> .  
<B> skos:broadMatch <C> .  
<A> skos:relatedMatch <C> .
```

10.6.3. Властивості співставлень і транзитивність

Тільки властивості співставлення SKOS, які оголошені як транзитивні є skos:exactMatch. Приклад втілення показано нижче:

Приклад 62 (втілення)

```
<A> skos:exactMatch <B> .  
<B> skos:exactMatch <C> .  
втілює  
<A> skos:exactMatch <C> .
```

Всі інші властивості співставлення SKOS не є транзитивними. Таким чином, втілення як показано на прикладі нижче **не** підтримується моделлю даних SKOS.

Приклад 63 (без втілення)

```
<A> skos:broadMatch <B> .  
<B> skos:broadMatch <C> .  
не втілює  
<A> skos:broadMatch <C> .
```

Приклад 64 (без втілення)

```
<A> skos:relatedMatch <B> .  
<B> skos:relatedMatch <C> .  
не втілює  
<A> skos:relatedMatch <C> .
```

Приклад 65 (без втілення)

```
<A> skos:closeMatch <B> .  
<B> skos:closeMatch <C> .  
не втілює  
<A> skos:closeMatch <C> .
```

10.6.4. Властивості співставлення і рефлексивність

Ні одна властивість співставлення SKOS не є ні рефлексивною, ні іррефлексивною.

Оскільки `skos:exactMatch`, `skos:broadMatch` і `skos:relatedMatch` є не іррефлексивними, приклад нижче, **відповідає** моделі даних SKOS.

Приклад 66 (відповідає)

```
<A> skos:exactMatch <A> .  
<B> skos:broadMatch <B> .  
<C> skos:relatedMatch <C> .
```

Тим не менше, див. також [розділ 8](#) про рефлексивність властивостей семантичних відносин SKOS.

10.6.5. Цикли і альтернативні шляхи, які залучають `Skos: broadMatch`

Немає формальних умов цілісності запобігання ні циклів, ні альтернативних шляхів у графі ієрархічних зв'язків співставлення.

В графі нижче є два цикли за участю `skos:broadMatch`. Ця графа **відповідає** моделі даних SKOS.

Приклад 67 (відповідає)

```
<A> skos:broadMatch <B> .  
<B> skos:broadMatch <A> .  
  
<X> skos:broadMatch <Y> .  
<Y> skos:broadMatch <Z> .  
<Z> skos:broadMatch <X> .
```

В графі нижче є два альтернативних шляхи, які залучають `skos:broadMatch`. Цей приклад **відповідає** моделі даних SKOS.

Приклад 68 (відповідає)

```
<A> skos:broadMatch <B> .  
<B> skos:broadMatch <C> .  
<A> skos:broadMatch <C> .
```

Див. [розділ 8](#) про цикли і альтернативні шляхи, які залучають `skos:broader`.

10.6.6. Цикли, які залучають `skos:exactMatch` і `skos:closeMatch`

Приклад 69 (втілення)

```
<A> skos:exactMatch <B>  
втілює  
<A> skos:exactMatch <A> .  
<A> skos:closeMatch <A> .
```

Із-за втілення, показаного вище (яка виникає в результаті поєднання [S42](#), [S44](#) і [S45](#)), додатки повинні бути в змозі працювати з циклами в `skos:exactMatch` і `skos:closeMatch`.

10.6.7. Під-властивість рядів, які залучають `skos:exactMatch`

Немає аксіом під-властивості систем в моделі даних Skos, які залучають властивості `skos:exactMatch` або `skos:closeMatch`. Тому втілення, які показано нижче, **не** підтримуються моделлю даних Skos.

Приклад 70 (без втілення)

```
<A> skos:exactMatch <B> .
```

```
<B> skos:broadMatch <C> .
```

не втілює

```
<A> skos:broadMatch <C> .
```

Приклад 71 (без втілення)

```
<A> skos:exactMatch <B> .
```

```
<B> skos:relatedMatch <C> .
```

не втілює

```
<A> skos:relatedMatch <C> .
```

Приклад 72 (без втілення)

```
<A> skos:closeMatch <B> .
```

```
<B> skos:broadMatch <C> .
```

не втілює

```
<A> skos:broadMatch <C> .
```

Приклад 73 (без втілення)

```
<A> skos:closeMatch <B> .
```

```
<B> skos:relatedMatch <C> .
```

не втілює

```
<A> skos:relatedMatch <C> .
```

10.6.8. skos:closeMatch, skos:exactMatch, owl:sameAs, owl:equivalentClass, owl:equivalentProperty

OWL надає три властивості, які можуть на перший погляд здатися схожими на skos:closeMatch або skos:exactMatch. owl:sameAs використовується, щоб зв'язати два індивіда в онтології, і вказує, що вони є однаковими індивідами (наприклад, однаковий ресурс) . owl:equivalentClass використовується для зв'язку між двома класами в онтології, і вказує, що ці класи мають однаковий клас розширень. owl:equivalentProperty використовується, щоб зв'язати дві властивості в онтології і вказує, що обидві властивості мають однакові властивості розширення.

skos:closeMatch і skos:exactMatch використовуються, щоб зв'язати концепти SKOS у різних схемах. Зв'язок skos:closeMatch показує, що два концепти є досить схожими, що вони можуть бути використані як взаємозамінні в **деяких** додатках інформаційного пошуку. Зв'язок skos:exactMatch вказує на високу ступінь впевненості, що два поняття можуть бути взаємозамінними в широкому спектрі додатків інформаційного пошуку.

owl:sameAs, owl:equivalentClass або owl:equivalentProperty як правило, недоречні для зв'язку концептів SKOS у різних концептуальних схемах, тому що наслідки, які слідують можуть бути небажаними.

Приклад нижче ілюструє деякі небажані втілення, які будуть слідувати від використання owl:sameAs .

Приклад 74 (втілення)

```
<A> rdf:type skos:Concept ;  
    skos:prefLabel "love"@en ;  
    skos:inScheme <MyScheme> .
```

```
<B> rdf:type skos:Concept ;  
    skos:prefLabel "adoration"@en ;
```

```
skos:inScheme <AnotherScheme> .
```

```
<A> owl:sameAs <B> .
```

втілює

```
<A>
```

```
skos:prefLabel "love"@en ;  
skos:prefLabel "adoration"@en ;  
skos:inScheme <MyScheme> ;  
skos:inScheme <AnotherScheme> .
```

```
<B>
```

```
skos:prefLabel "love"@en ;  
skos:prefLabel "adoration"@en ;  
skos:inScheme <MyScheme> ;  
skos:inScheme <AnotherScheme> .
```

У цьому прикладі, використання owl:sameAs для зв'язку двох концептів SKOS у різних концептуальних схемах фактично приведе до несумісності з моделлю даних Skos, тому що обидва <A> і тепер мають дві бажаних лексичних позначки на тій же мові. Хоча це не завжди так.

11. Список літератури

[AGROVOC]

[AGROVOC Thesaurus](http://www.fao.org/agrovoc), Food and Agriculture Organization of the United Nations (FAO). Наявний на <http://www.fao.org/agrovoc>

[BCP47]

[Tags for Identifying Languages](http://www.rfc-editor.org/rfc/bcp/bcp47.txt), A. Phillips and M. Davis, Editors, September 2006. Наявний на <http://www.rfc-editor.org/rfc/bcp/bcp47.txt>

[BS8723-2]

BS8723 Structured Vocabularies for Information Retrieval Part 2: Thesauri, British Standards Institution (BSI), 2005.

[BS8723-3]

BS8723 Structured Vocabularies for Information Retrieval Part 3: Vocabularies Other Than Thesauri, British Standards Institution (BSI), 2005.

[COOLURIS]

[Cool URIs for the Semantic Web](http://www.w3.org/TR/2008/NOTE-cooluris-20080331/), Leo Sauermann and Richard Cyganiak, Editors, W3C Interest Group Note, 31 March 2008, <http://www.w3.org/TR/2008/NOTE-cooluris-20080331/>. [Остання версія](http://www.w3.org/TR/cooluris/) доступна за HTTP : / / www.w3.org/TR/cooluris/

[ISO2788]

ISO 2788:1986 Documentation -- Guidelines for the establishment and development of monolingual thesauri, International Organization for Standardization (ISO), 1986.

[LCSH]

[Library of Congress Subject Headings](http://www.loc.gov/cds/lcsh.html), The Library of Congress Cataloging Distribution Service. Наявний на <http://www.loc.gov/cds/lcsh.html> і <http://id.loc.gov/>

[NTRIPLES]

[RDF Test Cases](http://www.w3.org/TR/2004/REC-rdf-testcases-20040210/), Jan Grant and Dave Beckett, Editors, W3C Recommendation, 10 February 2004, <http://www.w3.org/TR/2004/REC-rdf-testcases-20040210/>. [Остання версія](http://www.w3.org/TR/rdf-testcases/) доступна на [http:// www.w3.org/TR/rdf-testcases/](http://www.w3.org/TR/rdf-testcases/)

[OWL-GUIDE]

[OWL Web Ontology Language Guide](#), Michael K. Smith, Chris Welty and Deborah L. McGuinness, Editors, W3C Recommendation, 10 February 2004, <http://www.w3.org/TR/2004/REC-owl-guide-20040210/>. [Остання версія](#) доступна на <http://www.w3.org/TR/owl-guide/>

[OWL-REFERENCE]

[OWL Web Ontology Language Reference](#), Mike Dean and Guus Schreiber, Editors, W3C Recommendation, 10 February 2004, <http://www.w3.org/TR/2004/REC-owl-ref-20040210/>. [Остання версія](#) доступна за адресою <http://www.w3.org/TR/owl-ref/>

[OWL-SEMANTICS]

[OWL Web Ontology Language Semantics and Abstract Syntax](#), Peter F. Patel-Schneider, Patrick Hayes and Ian Horrocks, Editors, W3C Recommendation, 10 February 2004, <http://www.w3.org/TR/2004/REC-owl-semantics-20040210/>. [Остання версія](#) доступна на <http://www.w3.org/TR/owl-semantics/>

[RDF-CONCEPTS]

[Resource Description Framework \(RDF\): Concepts and Abstract Syntax](#), Graham Klyne and Jeremy J. Carroll, Editors, W3C Recommendation, 10 February 2004, <http://www.w3.org/TR/2004/REC-rdf-concepts-20040210/>. [Остання версія](#) доступна на <http://www.w3.org/TR/rdf-concepts/>

[RDF-PRIMER]

[RDF Primer](#), Frank Manola and Eric Miller, Editors, W3C Recommendation, 10 February 2004, <http://www.w3.org/TR/2004/REC-rdf-primer-20040210/>. [Остання версія](#) доступна на <http://www.w3.org/TR/rdf-primer/>

[RDF-SEMANTICS]

[RDF Semantics](#), Patrick Hayes, Editor, W3C Recommendation, 10 February 2004, <http://www.w3.org/TR/2004/REC-rdf-mt-20040210/>. [Остання версія](#) доступна на <http://www.w3.org/TR/RDF-MT/>

[RDF-XML]

[RDF/XML Syntax Specification \(Revised\)](#), Dave Beckett, Editor, W3C Recommendation, 10 February 2004, <http://www.w3.org/TR/2004/REC-rdf-syntax-grammar-20040210/>. [Остання версія](#) доступна на <http://www.w3.org/TR/rdf-syntax-grammar/>

[RDFS]

[RDF Vocabulary Description Language 1.0: RDF Schema](#), Dan Brickley and R. V. Guha, Editors, W3C Recommendation, 10 February 2004, <http://www.w3.org/TR/2004/REC-rdf-schema-20040210/>. [Остання версія](#) доступна на <http://www.w3.org/TR/rdf-schema/>

[RECIPES]

[Best Practice Recipes for Publishing RDF Vocabularies](#), Diego Berrueta and Jon Phipps, Editors, W3C Working Draft, 23 January 2008, <http://www.w3.org/TR/2008/WD-swbp-vocab-pub-20080123/>. [Останній Версія](#) доступна на <http://www.w3.org/TR/swbp-vocab-pub/>

[SKOS-HTML]

[SKOS Namespace Document - HTML Variant](#). [Остання версія](#) доступна на <http://www.w3.org/TR/skos-reference/skos.html>

[SKOS-PRIMER]

[SKOS Simple Knowledge Organization System Primer](#), Antoine Isaac and Ed Summers, Editors, W3C Working Group Note, 18 August 2009, <http://www.w3.org/TR/2009/NOTE-skos-primer-20090818/>. [Остання версія](#) доступна на <http://www.w3.org/TR/skos-primer>

[SKOS-RDF]

[SKOS Namespace Document - RDF/XML Variant](#). [Остання версія](#) доступна на <http://www.w3.org/TR/skos-reference/skos.rdf>

[SKOS-RDF-OWL1-DL]

[SKOS RDF Schema - OWL 1 DL Sub-set](http://www.w3.org/TR/skos-reference/skos-owl1-dl.rdf). [Остання версія](#) доступна на <http://www.w3.org/TR/skos-reference/skos-owl1-dl.rdf>

[SKOS-UCR]

[SKOS Use Cases and Requirements](http://www.w3.org/TR/2009/NOTE-skos-ucr-20090818), Antoine Isaac, Jon Phipps and Daniel Rubin, Editors, W3C Working Group Note, 18 August 2009, <http://www.w3.org/TR/2009/NOTE-skos-ucr-20090818>. [Остання версія](#) доступна на <http://www.w3.org/TR/skos-ucr>

[SKOS-XL-HTML]

[SKOS-XL Namespace Document - HTML Variant](http://www.w3.org/TR/skos-reference/skos-xl.html). [Остання версія](#) доступна на <http://www.w3.org/TR/skos-reference/skos-xl.html>

[SKOS-XL-RDF]

[SKOS-XL Namespace Document - RDF/XML Variant](http://www.w3.org/TR/skos-reference/skos-xl.rdf). [Остання версія](#) доступна на <http://www.w3.org/TR/skos-reference/skos-xl.rdf>

[SPARQL]

[SPARQL Query Language for RDF](http://www.w3.org/TR/2008/REC-rdf-sparql-query-20080115/), Eric Prud'hommeaux and Andy Seaborne, Editors, W3C Recommendation, 15 January 2008, <http://www.w3.org/TR/2008/REC-rdf-sparql-query-20080115/>. [Остання версія](#) доступна на <http://www.w3.org/TR/rdf-sparql-query/>

[SW]

[W3C Semantic Web Activity](http://www.w3.org/2001/sw/). Наявний на <http://www.w3.org/2001/sw/>

[SWBP-DATATYPES]

[XML Schema Datatypes in RDF and OWL](http://www.w3.org/TR/2006/NOTE-swbp-xsch-datatypes-20060314/), Jeremy J. Carroll and Jeff Z. Pan, Editors, W3C Working Group Note, 14 March 2006, <http://www.w3.org/TR/2006/NOTE-swbp-xsch-datatypes-20060314/>. [Остання версія](#) доступна на <http://www.w3.org/TR/swbp-xsch-datatypes/>

[TURTLE]

[Turtle - Terse RDF Triple Language](http://www.w3.org/TeamSubmission/2008/SUBM-turtle-20080114/), David Beckett and Tim Berners-Lee, W3C Team Submission, 14 January 2008, <http://www.w3.org/TeamSubmission/2008/SUBM-turtle-20080114/>. [Остання версія](#) доступна на <http://www.w3.org/TeamSubmission/turtle/>

[[WEBARCH]

[Architecture of the World Wide Web, Volume One](http://www.w3.org/TR/2004/REC-webarch-20041215/), Ian Jacobs and Norman Walsh, Editors, W3C Recommendation, 15 December 2004, <http://www.w3.org/TR/2004/REC-webarch-20041215/>. [Остання версія](#) доступна на <http://www.w3.org/TR/webarch/>

[XML-SCHEMA]

[XML Schema Part 2: Datatypes Second Edition](http://www.w3.org/TR/2004/REC-xmlschema-2-20041028/), Paul V. Biron and Ashok Malhotra, Editors, W3C Recommendation, 28 October 2004, <http://www.w3.org/TR/2004/REC-xmlschema-2-20041028/>. [Остання версія](#) доступна на <http://www.w3.org/TR/xmlschema-2/>

12. Подяки

Цей документ є результатом широкого обговорення в рамках [W3C Semantic Web Deployment Working Group](#). Документ спирається на досвід попередніх груп і проєктів, у тому числі [SWAD-Europe](#) і [W3C Semantic Web Best Practices and Deployment Working Group](#). Члени [public-esw-thes](#) розсилки консорціуму W3C також внесли цінний вклад.

Додаток А. Властивості і класи SKOS

А.1. Класи в моделі даних SKOS

<u>skos:Collection</u>	
URI:	http://www.w3.org/2004/02/skos/core#Collection
Визначення:	Розділ 9. Концептуальні колекції
Лейбл:	<i>Collection</i>
Непересічні класи:	skos:Concept skos:ConceptScheme
<u>skos:Concept</u>	
URI:	http://www.w3.org/2004/02/skos/core#Concept
Визначення:	Розділ 3. Клас skos:Concept
Лейбл:	<i>Concept</i>
Непересічні класи:	skos:Collection skos:ConceptScheme
<u>skos:ConceptScheme</u>	
URI:	http://www.w3.org/2004/02/skos/core#ConceptScheme
Визначення:	Розділ 4. Концептуальні схеми
Лейбл:	<i>Concept Scheme</i>
Непересічні класи:	skos:Collection skos:Concept
<u>skos:OrderedCollection</u>	
URI:	http://www.w3.org/2004/02/skos/core#OrderedCollection
Визначення:	Розділ 9. Концептуальні колекції
Лейбл:	<i>Ordered Collection</i>

Над-класи:	skos:Collection

A.2. Властивості в моделі даних SKOS

skos:altLabel	
URI:	http://www.w3.org/2004/02/skos/core#altLabel
Визначення:	Розділ 5. Лексичні позначки
Лейбл:	<i>alternative label</i>
Над-властивості:	http://www.w3.org/2000/01/rdf-schema#label
skos:broadMatch	
URI:	http://www.w3.org/2004/02/skos/core#broadMatch
Визначення:	Розділ 10. Властивості співставлення
Лейбл:	<i>has broader match</i>
Над-властивості:	skos:broader skos:mappingRelation
Зворотні:	skos:narrowMatch
skos:broader	
URI:	http://www.w3.org/2004/02/skos/core#broader
Визначення:	Розділ 8. Семантичні відносини
Лейбл:	<i>has broader</i>
Над-властивості:	skos:broaderTransitive
Зворотні:	skos:narrower
skos:broaderTransitive	

URI:	http://www.w3.org/2004/02/skos/core#broaderTransitive
Визначення:	Розділ 8. Семантичні відносини
Лейбл:	<i>has broader transitive</i>
Над-властивості:	skos:semanticRelation
Зворотні:	skos:narrowerTransitive
Інші характеристики:	Транзитивні
skos:changeNote	
URI:	http://www.w3.org/2004/02/skos/core#changeNote
Визначення:	Розділ 7. Властивості документації
Лейбл:	<i>change note</i>
Над-властивості:	skos:note
skos:closeMatch	
URI:	http://www.w3.org/2004/02/skos/core#closeMatch
Визначення:	Розділ 10. Властивості співставлення
Лейбл:	<i>has close match</i>
Над-властивості:	skos:mappingRelation
Інші характеристики:	Симетричні
skos:definition	
URI:	http://www.w3.org/2004/02/skos/core#definition
Визначення:	Розділ 7. Властивості документації
Лейбл:	<i>definition</i>

Над-властивості:	skos:note
skos:editorialNote	
URI:	http://www.w3.org/2004/02/skos/core#editorialNote
Визначення:	Розділ 7. Властивості документації
Лейбл:	<i>editorial note</i>
Над-властивості:	skos:note
skos:exactMatch	
URI:	http://www.w3.org/2004/02/skos/core#exactMatch
Визначення:	Розділ 10. Властивості співставлення
Лейбл:	<i>has exact match</i>
Над-властивості:	skos:closeMatch
Інші характеристики:	Транзитивні Симетричні
skos:example	
URI:	http://www.w3.org/2004/02/skos/core#example
Визначення:	Розділ 7. Властивості документації
Лейбл:	<i>example</i>
Над-властивості:	skos:note
skos:hasTopConcept	
URI:	http://www.w3.org/2004/02/skos/core#hasTopConcept
Визначення:	Розділ 4. Концептуальні схеми
Лейбл:	<i>label</i>

Домен:	skos:ConceptScheme
Діапазон:	skos:Concept
Зворотні:	skos:topConceptOf
skos:hiddenLabel	
URI:	http://www.w3.org/2004/02/skos/core#hiddenLabel
Визначення:	Розділ 5. Лексичні позначки
Лейбл:	<i>hidden label</i>
Над-властивості:	http://www.w3.org/2000/01/rdf-schema#label
skos:historyNote	
URI:	http://www.w3.org/2004/02/skos/core#historyNote
Визначення:	Розділ 7. Властивості документації
Лейбл:	<i>history note</i>
Над-властивості:	skos:note
skos:inScheme	
URI:	http://www.w3.org/2004/02/skos/core#inScheme
Визначення:	Розділ 4. Концептуальні схеми
Лейбл:	<i>is in scheme</i>
Діапазон:	skos:ConceptScheme
skos:mappingRelation	
URI:	http://www.w3.org/2004/02/skos/core#mappingRelation
Визначення:	Розділ 10. Властивості співставлення
Лейбл:	<i>is in mapping relation with</i>

Над-властивості:	skos:semanticRelation
skos:member	
URI:	http://www.w3.org/2004/02/skos/core#member
Визначення:	Розділ 9. Концептуальні колекції
Лейбл:	<i>has member</i>
Домен:	skos:Collection
Діапазон:	з'єднання skos:Concept і skos:Collection
skos:memberList	
URI:	http://www.w3.org/2004/02/skos/core#memberList
Визначення:	Розділ 9. Концептуальні колекції
Лейбл:	<i>has member list</i>
Домен:	skos:OrderedCollection
Діапазон:	http://www.w3.org/1999/02/22-rdf-syntax-ns#List
Інші характеристики:	Функціональні
skos:narrowMatch	
URI:	http://www.w3.org/2004/02/skos/core#narrowMatch
Визначення:	Розділ 10. Властивості співставлення
Лейбл:	<i>has narrower match</i>
Над-властивості:	skos:mappingRelation skos:narrower
Зворотні:	skos:broadMatch
skos:narrower	
URI:	http://www.w3.org/2004/02/skos/core#narrower

Визначення:	Розділ 8. Семантичні відносини
Лейбл:	<i>has narrower</i>
Над-властивості:	skos:narrowerTransitive
Зворотні:	skos:broader
skos:narrowerTransitive	
URI:	http://www.w3.org/2004/02/skos/core#narrowerTransitive
Визначення:	Розділ 8. Семантичні відносини
Лейбл:	<i>has narrower transitive</i>
Над-властивості:	skos:semanticRelation
Зворотні:	skos:broaderTransitive
Інші характеристики:	Транзитивні
skos:notation	
URI:	http://www.w3.org/2004/02/skos/core#notation
Визначення:	Розділ 6. Нотатки
Лейбл:	<i>notation</i>
skos:note	
URI:	http://www.w3.org/2004/02/skos/core#note
Визначення:	Розділ 7. Властивості документації
Лейбл:	<i>note</i>
skos:prefLabel	
URI:	http://www.w3.org/2004/02/skos/core#prefLabel
Визначення:	Розділ 5. Лексичні позначки

Лейбл:	<i>preferred label</i>
Над-властивості:	http://www.w3.org/2000/01/rdf-schema#label
skos:related	
URI:	http://www.w3.org/2004/02/skos/core#related
Визначення:	Розділ 8. Семантичні відносини
Лейбл:	<i>has related</i>
Над-властивості:	skos:semanticRelation
Інші характеристики:	Симетричні
skos:relatedMatch	
URI:	http://www.w3.org/2004/02/skos/core#relatedMatch
Визначення:	Розділ 10. Властивості співставлення
Лейбл:	<i>has related match</i>
Над-властивості:	skos:mappingRelation skos:related
Інші характеристики:	Симетричні
skos:scopeNote	
URI:	http://www.w3.org/2004/02/skos/core#scopeNote
Визначення:	Розділ 7. Властивості документації
Лейбл:	<i>scope note</i>
Над-властивості:	skos:note
skos:semanticRelation	
URI:	http://www.w3.org/2004/02/skos/core#semanticRelation
Визначення:	Розділ 8. Семантичні відносини

Лейбл:	<i>is in semantic relation with</i>
Домен:	skos:Concept
Діапазон:	skos:Concept
skos:topConceptOf	
URI:	http://www.w3.org/2004/02/skos/core#topConceptOf
Визначення:	Розділ 4. Концептуальні схеми
Лейбл:	<i>is top concept in scheme</i>
Над-властивості:	skos:inScheme
Зворотні:	skos:hasTopConcept

Додаток В. SKOS eXtension for Labels (SKOS-XL)

Цей додаток визначає **додаткове** розширення простої системи організації знання, що отримало назву SKOS eXtension for Labels (SKOS-XL). Це розширення забезпечує додаткову підтримку для виявлення, опису та зв'язку лексичних форм.

Особливий клас лексичних форм, який називається `skosxl:Label` є визначеним. Кожен екземпляр цього класу має одну форму RDF текстового літерала, але два екземпляри цього класу не обов'язково повинні бути однаковими, якщо вони мають однакову форму літерала.

Три властивості позначки `skosxl:prefLabel`, `skosxl:altLabel` і `skosxl:hiddenLabel` є визначеними. Ці властивості використовуються для позначення SKOS концептів з екземплярами `skosxl:Label`, і є розширеними аналогами властивостей однакових локальних назв, визначених в SKOS (`skos:prefLabel`, `skos:altLabel` і `skos:hiddenLabel`).

Модель даних SKOS також визначає властивість `skosxl:labelRelation`. Ця властивість може бути використана для ствердження прямих (двійкових) зв'язків між екземплярами `skosxl:Label`. Вона призначена перш за все для розширення точок, які будуть уточнені для більш конкретних видів зв'язку. Не надаються ніякі вбудовані уточнення `skosxl:labelRelation`, хоча і наводяться деякі приклади того, як це може бути зроблено або дано.

В.1. SKOS-XL простір імен і словник

URI для простору імен SKOS-XL є:

- <http://www.w3.org/2008/05/skos-xl#>

Тут префікс skosxl: використовується як аббревіатура для URI простору імен SKOS-XL. Словник SKOS-XL є сукупністю URI, поданих в лівій колонці таблиці нижче.

Таблиця 2. Словник SKOS-XL

URI	Визначається в (розділ цього додатку)
skosxl:Label	Клас skosxl:Label
skosxl:literalForm	Клас skosxl:Label
skosxl:prefLabel	Привілейовані, альтернативні і приховані skosxl:Labels
skosxl:altLabel	Привілейовані, альтернативні і приховані skosxl:Labels
skosxl:hiddenLabel	Привілейовані, альтернативні і приховані skosxl:Labels
skosxl:labelRelation	Зв'язок між skosxl:Labels

Тут "словник SKOS-XL" відноситься до об'єднання словника SKOS і словника SKOS-XL.

Тут "модель даних XL" відноситься до визначень класу і властивостей, які представлені тільки у цьому додатку. "Модель даних SKOS + XL" відноситься до об'єднання моделі даних, представлених в розділах 3-10 вище, і моделі даних XL.

B.2. Клас skosxl:Label

B.2.1. Преамбула

Клас skosxl:Label - це особливий клас лексичних форм.

Екземпляром класу skosxl:Label є ресурс, який може бути названий з URI.

Екземпляр класу skosxl:Label має просту форму літерала. Ця форма літерала - це простий літерал RDF (який є рядком символів в кодуванні UNICODE і додатковим мовним тегом [[RDF-CONCEPTS](#)]). Властивість skosxl:literalForm використовується, щоб дати форму літерала skosxl:Label.

Якщо два екземпляри класу skosxl:Label мають однакові форми літерала, вони **не** обов'язково є однаковими ресурсами.

B.2.2. Визначення класів і властивостей

S47	skosxl:Label є екземпляром owl:Class.
S48	skosxl:Label не сумісний з skos:Concept, skos:ConceptScheme і skos:Collection.
S49	skosxl:literalForm є екземпляром owl:DatatypeProperty.

S50	rdfs:domain з skosxl:literalForm є класом skosxl:Label.
S51	rdfs:range з skosxl:literalForm є класом RDF текстового літерала.
S52	skosxl:Label є під-класом обмежень на skosxl:literalForm потужністю точно 1.

В.2.3. Приклади

Наведений нижче приклад описує skosxl:Label, який названо з URI <http://example.com/ns/A>, формою літерала англійською мовою.

Приклад 75 (відповідає)

```
<A> rdf:type skosxl:Label ; skosxl:literalForm "love"@en .
```

Чотири приклади, наведені нижче **не відповідають** моделі даних XL, тому що skosxl:Label описується двома різними формами літералів.

Приклад 76 (не відповідає)

```
<B> rdf:type skosxl:Label ; skosxl:literalForm "love" ; skosxl:literalForm "adoration" .
```

Приклад 77 (не відповідає)

```
<B> rdf:type skosxl:Label ; skosxl:literalForm "love"@en ; skosxl:literalForm "love"@fr .
```

Приклад 78 (не відповідає)

```
<B> rdf:type skosxl:Label ; skosxl:literalForm "love"@en-GB ; skosxl:literalForm "love"@en-US .
```

Приклад 79 (не відповідає)

```
<B> rdf:type skosxl:Label ; skosxl:literalForm "東"@ja-Hani ; skosxl:literalForm "ひがし"@ja-Hira .
```

В.2.4. Нотатки

В.2.4.1. Ідентичність та втілення

Як зазначалося вище, кожен екземпляр класу skosxl:Label має **одну і тільки одну форму літерала**. Іншими словами, існує функція співставлення класу розширення skosxl:Label та сукупності RDF текстових літералів. Ця функція визначена властивістю розширення skosxl:literalForm. Слід звернути особливу увагу на два факти про цю функцію.

По-перше, функція **не** є ін'єктивною. Іншими словами, **не** існує співставлення один-до-одного skosxl:Label та сукупності RDF текстових літералів (насправді це багато-до-одного). Це означає, що два екземпляри skosxl:Label, які мають однакову форму літерала **не обов'язково** є однаковими індивідами. Так, наприклад, втілення, яке проілюстровано нижче **не** підтримується моделлю даних XL.

Приклад 80 (без втілення)

```
<A> skosxl:literalForm "love"@en .
```

```
<B> skosxl:literalForm "love"@en .
```

Не втілює

```
<A> owl:sameAs <B> .
```

По-друге, дана функція не є сюр'єктивною. Іншими словами, для даного простого літерала I, не може бути ніяких прикладів skosxl:Label з формою літерала 1.

В.2.4.2. Склад концептуальних схем

До складу екземпляра skosxl:Label в рамках концептуальної схеми SKOS можуть бути зараховані властивості skos:inScheme.

Приклад 81 (відповідає)

```
<A> rdf:type skosxl:Label ; skosxl:literalForm "love"@en ; skos:inScheme <MyScheme> .
```

В.3. Привілейовані, альтернативні і приховані

В.3.1. Преамбула

Три властивості skosxl:prefLabel, skosxl:altLabel і skosxl:hiddenLabel використовуються для подачі привілейованих, альтернативних і прихованих позначок відповідно до ресурсу, де ці позначки є екземплярами класу skosxl:Label. Ці властивості є аналогічними властивостям однакових імен, визначених в словнику Skos, і є логічними залежностями між цими двома групами властивостей, визначених нижче.

В.3.2. Визначення класів і властивостей

S53	skosxl:prefLabel, skosxl:altLabel і skosxl:hiddenLabel є екземплярами owl:ObjectProperty.
S54	rdfs:range КОЖНОГО з skosxl:prefLabel, skosxl:altLabel і skosxl:hiddenLabel є класом skosxl:Label.
S55	Властивість ряду (skosxl:prefLabel, skosxl:literalForm) є під-властивістю skos:prefLabel.
S56	Властивість ряду (skosxl:altLabel, skosxl:literalForm) є під-властивістю skos:altLabel.
S57	Властивість ряду (skosxl:hiddenLabel, skosxl:literalForm) є під-властивістю skos:hiddenLabel.
S58	skosxl:prefLabel, skosxl:altLabel і skosxl:hiddenLabel є властивостями, що попарно непересічні.

В.3.3. Приклади

Наведений нижче приклад ілюструє використання всіх трьох властивостей позначок XL і відповідає моделі даних SKOS + XL.

Приклад 82 (відповідає)

```
<Love>  
skosxl:prefLabel <A> ;  
skosxl:altLabel <B> ;
```

```
skosxl:hiddenLabel <C> .

<A> rdf:type skosxl:Label ;
skosxl:literalForm "love"@en .

<B> rdf:type skosxl:Label ;
skosxl:literalForm "adoration"@en .

<C> rdf:type skosxl:Label ;
skosxl:literalForm "luv"@en .
```

В.3.4. Нотатки

В.3.4.1. *Dumbing-Down* до SKOS лексичних позначок

Під-властивість ряду аксіом [S55](#), [S56](#) і [S57](#) підтримує *dumbing-down* для позначок XL. Це показано на прикладі нижче.

Приклад 83 (втілює)

```
<Love>
skosxl:prefLabel <A> ;
skosxl:altLabel <B> ;
skosxl:hiddenLabel <C> .

<A> rdf:type skosxl:Label ;
skosxl:literalForm "love"@en .

<B> rdf:type skosxl:Label ;
skosxl:literalForm "adoration"@en .

<C> rdf:type skosxl:Label ;
skosxl:literalForm "luv"@en .
```

втілює

```
<Love>
skos:prefLabel "love"@en ;
skos:altLabel "adoration"@en ;
skos:hiddenLabel "luv"@en .
```

В.3.4.2. Цілісність позначок SKOS + XL

У [розділі 5](#), дві умови цілісності були визначені на основні властивостей позначок SKOS. По-перше, властивості `skos:prefLabel`, `skos:altLabel` і `skos:hiddenLabel` попарно непересічні. По-друге, ресурс не має більше одного значення `skos:prefLabel` для кожної мови. Завдяки під-властивостям ряду аксіом, визначеним вище, наступні чотири приклади, у той час як відносно відповідають моделі даних XL поодиноці, **не узгоджуються** з моделлю даних SKOS + XL.

Приклад 84 (не відповідає)

Two different preferred labels in the same language

```
<Love> skosxl:prefLabel <A> ; skosxl:prefLabel <B> .  
<A> skosxl:literalForm "love"@en .  
<B> skosxl:literalForm "adoration"@en .
```

Приклад 85 (не відповідає)

Clash between preferred and alternative labels

```
<Love> skosxl:prefLabel <A> ; skosxl:altLabel <B> .  
<A> skosxl:literalForm "love"@en .  
<B> skosxl:literalForm "love"@en .
```

Приклад 86 (не відповідає)

Clash between alternative and hidden labels

```
<Love> skosxl:altLabel <A> ; skosxl:hiddenLabel <B> .  
<A> skosxl:literalForm "love"@en .  
<B> skosxl:literalForm "love"@en .
```

Приклад 87 (не відповідає)

Clash between preferred and hidden labels

```
<Love> skosxl:prefLabel <A> ; skosxl:hiddenLabel <B> .  
<A> skosxl:literalForm "love"@en .  
<B> skosxl:literalForm "love"@en .
```

B.4. Зв'язки між skosxl:Labels_

B.4.1. Преамбула

Цей розділ визначає шаблон для подання бінарних зв'язків між екземплярами класу skosxl:Label.

Зверніть увагу, що словник, визначений в цьому розділі, не призначений для безпосереднього використання, а, скоріше, як точка розширення, яка може бути уточнена для більш специфічних позначок.

B.4.2. Визначення класів і властивостей

S59	skosxl:labelRelation є екземплярами owl:ObjectProperty.
S60	rdfs:domain з skosxl:labelRelation є класом skosxl:Label.
S61	rdfs:range з skosxl:labelRelation є класом skosxl:Label.
S62	skosxl:labelRelation є екземплярами owl:SymmetricProperty.

В.4.3. Приклади

Приклад нижче ілюструє зв'язок між двома екземплярами класу `skosxl:Label`

Приклад 88 (відповідає)

```
<A> rdf:type skosxl:Label ; skosxl:literalForm "love" .  
<B> rdf:type skosxl:Label ; skosxl:literalForm "adoration" .  
<A> skosxl:labelRelation <B> .
```

В.4.4. Нотатки

В.4.4.1. Уточнення цієї схеми

Як уже згадувалося вище, властивість `skosxl:labelRelation` виступає в якості точки розширення, яка може бути уточнена для більш специфічних позначок.

У прикладі, наведеному нижче, третя сторона уточнює властивість `skos:labelRelation`, щоб висловити скорочені відносини, і використовується, щоб висловити той факт, що "FAO" є скороченням від "Food and Agriculture Organization".

Приклад 89 (відповідає)

```
# First define an extension to skosxl:labelRelation  
ex:acronym rdfs:subPropertyOf skosxl:labelRelation .  
  
# Now use it  
<A> rdf:type skosxl:Label ; skosxl:literalForm "FAO"@en .  
<B> rdf:type skosxl:Label ; skosxl:literalForm "Food and Agriculture Organization"@en .  
<B> ex:acronym <A> .
```

Зверніть увагу, що під-властивість симетричної властивості - не обов'язково симетрична.

В.5. SKOS-XL схема огляду

У таблиці нижче наводиться огляд лексики SKOS-XL.

В.5.1 Класи у моделі даних SKOS-XL

<u>skosxl:Label</u>	
URI:	http://www.w3.org/2008/05/skos-xl#Label
Визначення:	Розділ В.2. Клас <code>skosxl:Label</code>
Лейбл:	<i>Label</i>

Над-класи:	Обмеження на skosxl:literalForm потужності рівно 1
Непересічні класи:	skos:Collection skos:Concept skos:ConceptScheme

В.5.2.Властивості в моделі даних SKOS-XL

skosxl:altLabel	
URI:	http://www.w3.org/2008/05/skos-xl#altLabel
Визначення:	Розділ В.3. Привілейовані, альтернативні і приховані skosxl:Labels
Лейбл:	<i>alternative label</i>
Діапазон:	skosxl:Label
skosxl:hiddenLabel	
URI:	http://www.w3.org/2008/05/skos-xl#hiddenLabel
Визначення:	Розділ В.3. Привілейовані, альтернативні і приховані skosxl:Labels
Лейбл:	<i>hidden label</i>
Діапазон:	skosxl:Label
skosxl:labelRelation	
URI:	http://www.w3.org/2008/05/skos-xl#labelRelation
Визначення:	Розділ В.4. Зв'язок між skosxl:Labels
Лейбл:	<i>label relation</i>
Домен:	skosxl:Label
Діапазон:	skosxl:Label
Інші характеристики:	симетрична

<u>skosxl:literalForm</u>	
URI:	http://www.w3.org/2008/05/skos-xl#literalForm
Визначення:	Розділ В.2. Клас skosxl:Label
Лейбл:	<i>literal form</i>
Домен:	skosxl:Label
<u>skosxl:prefLabel</u>	
URI:	http://www.w3.org/2008/05/skos-xl#prefLabel
Визначення:	Розділ В.3. Привілейовані, альтернативні і приховані skosxl:Labels
Лейбл:	<i>preferred label</i>
Діапазон:	skosxl:Label

Додаток С. Документи простору імен SKOS і SKOS-XL

Following Architecture World Wide Web, Volume One [\[WEBARCH\]](#). "документи простору імен" є "інформаційний ресурс, що містить корисну інформацію і може бути використаний для машин та / або людини".

Словник SKOS є концептуальним ресурсом, який визначений простором імен URI <http://www.w3.org/TR/skos-reference/>. Нормативне визначення словника SKOS знаходиться в посиланні SKOS(цей документ).

Наступні документи простору імен забезпечують альтернативне представлення словника SKOS:

С.1. SKOS документи простору імен - HTML варіант (нормативний)

Словник SKOS наводиться в документі простору імен SKOS - варіант HTML [\[SKOS-HTML\]](#), який подається з простору імен URI <http://www.w3.org/2004/02/skos/core#> через узгодження змісту, використовуючи Recipe 3 of "Best Practice Recipes for Publishing Vocabularies" [\[RECIPES\]](#). Клієнти, що вимагають HTML або XHTML повинні включати відповідні "Асерти" у заголовку запиту HTTP. З іншого боку, SKOS документи

простору імен - варіант HTML можна посилатися безпосередньо пославшись на його URI: <http://www.w3.org/TR/skos-reference/skos.html>

SKOS документи простору імен - варіант HTML повторює [Додатк А. SKOS властивості і класи](#) цього документа.

С.2. SKOS документи простору імен - RDF / XML варіант (нормативний)

SKOS документи простору імен - варіант RDF / XML [\[SKOS-RDF\]](#) висловлює лексику SKOS і моделі даних (наскільки можливо) як трійки RDF. Він подається через узгодження змісту, використовуючи Recipe 3 of "Best Practice Recipes for Publishing Vocabularies" [\[RECIPES\]](#). Клієнти, що вимагають RDF / XML повинні включати відповідні "Асерт" у заголовку запиту HTTP. Крім того, схема RDF може посилатися безпосередньо, наводячи свої URI: <http://www.w3.org/TR/skos-reference/skos.rdf>

Неможливо визначити всі твердження моделі даних SKOS як RDF трійки, тому ця схема утворює нормативні підмножини посилання SKOS. Схема RDF визначається як онтологія OWL Full [\[OWL-SEMANTICS\]](#) [\[OWL-REFERENCE\]](#). Словники SKOS можуть бути визначені як екземпляри цієї онтології.

С.3. SKOS RDF схема - OWL 1DL підмножина (інформативне)

Для зручності додатків, які хочуть працювати в рамках обмежень OWL DL, схема SKOS RDF - підмножина OWL 1 DL [\[SKOS-RDF-OWL1-DL\]](#) забезпечує модифікацію інформативних схем, яка відповідає цим обмеженням

SKOS підмножина OWL 1 DL може бути отримати за посиланням своєї URI: <http://www.w3.org/TR/skos-reference/skos-owl1-dl.rdf>

С.4. SKOS-XL документи простору імен - HTML варіант (нормативний)

Словник SKOS-XL приводиться в SKOS-XL документі простору імен - Варіант HTML [\[SKOS-XL-HTML\]](#), який подається з простору імен URI <http://www.w3.org/2008/05/skos-xl#> через узгодження змісту, використовуючи Recipe 3 of "Best Practice Recipes for Publishing Vocabularies" [\[RECIPES\]](#). Клієнти, що вимагають HTML або XHTML повинні включати відповідні "Асерт" у заголовку запиту HTTP. З іншого боку, документи простору імен SKOS-XL - варіант HTML можуть посилатися безпосередньо пославшись на його URI: <http://www.w3.org/TR/skos-reference/skos-xl.html>

Варіант HTML документу простору імен SKOS-XL перегукується з [Додаток В.5 SKOS-XL схема огляду](#) цього документа.

С.5. SKOS-XL документ простору імен - RDF / XML варіант (нормативний)

Цей документ схеми RDF висловлює словник SKOS і модель даних (наскільки можливо), як RDF трійки. Він подається з простору імен URI <http://www.w3.org/2008/05/skos-xl#> через узгодження змісту, використовуючи Recipe 3 of "Best Practice Recipes for Publishing Vocabularies" [\[RECIPES\]](#). Клієнти, що вимагають RDF / XML повинні включати відповідні "Асерт" у заголовку запиту HTTP. Крім того, схема RDF може посилатися безпосередньо, наводячи свої URI: <http://www.w3.org/TR/skos-reference/skos-xl.rdf>

Додаток D. Простір імен SKOS: історичні відомості

Схема SKOS визначає словник використовуючи простір імен [http://www.w3.org/2004/02/skos/core #](http://www.w3.org/2004/02/skos/core#). Цей простір імен був використаний для визначення первісної схеми SKOS, що стала відправною точкою для цієї рекомендації. У результаті цього, елементи, присутні в попередніх версіях машиночитаних схем були видалені з поточної версії. У ряді випадків, визначення або семантика елементів у схемі не змінилася.

Збереження існуючого простору імен SKOS дозволяє уникнути деяких проблем з існуючими KOS, які вже використовують схему SKOS. Користувачі повинні бути в курсі змін у семантиці skos:broader (і skos:narrower), які *можуть вплинути* на додатки SKOS.

Там, де елементи були видалені, не були висловлені ніякі несхвалення в схемі. Історичні версії схеми Skos отримані з [SKOS Web site "version history" page](#), і ті елементи, які були вилучені з останньої версії словника перераховані нижче.

- skos:symbol
- skos:prefSymbol
- skos:altSymbol
- skos:CollectableProperty
- skos:subject
- skos:isSubjectOf
- skos:primarySubject
- skos:isPrimarySubjectOf
- skos:subjectIndicator

У випадку skos:broader і skos:narrower, семантики елементів словника були змінені - ці властивості не оголошуються транзитивними. Таким чином, подальші втілення не виконується.

Приклад 90 (без втілення)

```
<A> skos:broader <B> .  
<B> skos:broader <C> .  
Не втілює  
<A> skos:broader <C> .
```

Транзитивна над-властивість skos:broader - skos:broaderTransitive – забезпечується для пошуку через транзитивні закриття відносин skos:broader. Аналогічна властивість - skos:narrowerTransitive - забезпечується для пошуку через транзитивні закриття skos:narrower .

