

# ПРЕДСТАВЛЕННЯ ТА ВІДОБРАЖЕННЯ БІБЛІОТЕЧНИХ ПРЕДМЕТНИХ КЛАСИФІКАЦІЙ ЗА ДОПОМОГОЮ ІНСТРУМЕНТІВ ОНТОЛОГІЙ

*О.М. Овдій, Г.Ю. Проскудіна*

Інститут програмних систем НАН України  
03187, Київ, проспект Академіка Глушкова, 40,  
т.: (044)526 6033; e-mail: gupros@isofts.kiev.ua

У роботі розглянуто предметні онтології (класифікації), що застосовуються у середовищі цифрових бібліотек. На прикладі комп'ютерної системи класифікації ACM наводяться можливі варіанти представлення на основі сімейства стандартів W3C у форматі XML, RDFS, OWL. А також описано досвід роботи по відображенню бібліотечних предметних класифікацій за допомогою інструментів онтологій, з метою вирішення задач інтероперабельності.

The paper describes subject-based ontologies (classifications) used in digital library environments. Possible representations of this classifications based on family of standards W3C in XML, RDFS, OWL formats are illustrated with an example ACM Computing Classification System. Besides, we describe our experience in classifications/ontologies mapping using ontology tools in order to solve interoperability problems.

## Вступ

Основне питання при побудові цифрових бібліотек полягає в тому, як організувати великі кількості інформації так, щоб користувачі могли знайти те, що їм потрібно. Описуючи ресурси, бібліотекарі розробили складні схеми класифікації й правила каталогізації для створення метаданих, що є головним елементом навколо якого організована будь-яка бібліотека. Для забезпечення інтероперабельності дані повинні бути представлені у вигляді придатному для машинної обробки, з цією метою використовуються мови онтологій. Крім того якщо дві бібліотеки використовують різні системи класифікації, як забезпечити інтероперабельність, ефективний обмін інформацією та її пошук? Відповіддю на це питання є відображення, вирівнювання та об'єднання онтологій, у нашому випадку бібліотечних класифікацій [1–2]. Фактично будь-яке застосування, що включає декілька онтологій повинно встановлювати семантичне відображення між ними. Незважаючи на своє поширення на сьогодні відображення онтологій у більшості своїй виконується вручну і є трудозатратним та піддатим до помилок процесом. У даній роботі процес відображення класифікацій розглянуто на прикладі двох бібліотечних класифікацій ACM та UDC. У рамках цієї роботи комп'ютерна система класифікація ACM перекладена на російську та українську мови.

Робота написана таким чином. Розділ 1 описує онтології, ACM й UDC, які приймали участь у відображенні, а також містить стислий опис інших широко використовуваних бібліотечних класифікацій. У розділі 2 описані мови представлення онтологій та наводяться приклади можливих варіантів їх представлення, а також використання інструменту Protégé як можливого інструменту здійснення такого представлення. Розділ 3 містить опис інструментів відображення онтологій. У розділі 4 наведено приклад відображення онтологій. Висновки наведені у розділі 5.

## 1. Бібліотечні каталоги

Багато бібліотечних застосувань ґрунтуються на публікації електронних предметних (тематичних) каталогів, що описують бібліотечні ресурси і дозволяють користувачам вибрати те, що їм потрібно. У даному розділі наведений опис класифікацій ACM та UDC, а також стислий опис інших широко використовуваних бібліотечних класифікацій.

**Комп'ютерна система класифікації ACM (CCS – Computing Classification System) перша система класифікації ACM для обчислювальної галузі була видана в 1964 р. [3–5]. У 1982 р. ACM видав цілком нову систему. Нові версії, засновані на системі 1982 року, з'явилися в 1983 р., 1987 р., 1991 й 1998 рр.**

Комп'ютерна система класифікації ACM (Association for Computing Machinery – Асоціації комп'ютерної техніки) стала стандартом ідентифікації й категоризації літератури по обчислювальній техніці, а також в галузі комп'ютерних наук. Повна схема класифікації є чотирьох рівневою ієрархічною таксономією (три закодованих рівні й четвертий незакодований), що містить загальні терміни й неявні предметні описувачі. Перший рівень є незмінним та включає 11 основних розділів:

- A. Загальна Література (General Literature).
- B. Апаратне забезпечення (Hardware).
- C. Устрій комп'ютерних систем (Computer Systems Organization).

- D. Програмне забезпечення (Software).
- E. Дані (Data).
- F. Теорія обчислень (Theory of Computation).
- G. Математика обчислень (Mathematics of Computing).
- H. Інформаційні Системи (Information Systems).
- I. Обчислювальні Методології (Computing Methodologies).
- J. Прикладні програми (Computer Applications).
- K. Середовища Обчислень (Computing Milieux).

Кількість вузлів на першому, другому, третьому та четвертому рівнях (версія 1998 р.) приблизно 11, 81, 277 та 1473. Терміни 2 та 3 рівнів іноді можуть мати перехресні посилання, відображаючи взаємозв'язки.

*Десяткова класифікація Д'юї* (DCC – Dewey Decimal Classification) найбільш широко застосовувана бібліотечна система класифікації у світі [6], де всі області знань діляться на 10 класів, а всередині кожного класу виділяють десяткові підкласи, розділи та підрозділи. У цей час вона використовується в 135 країнах і переведена на 30 мов. У США її використовують 95 % усіх публічних і шкільних бібліотек і величезне число бібліотек коледжів, університетів і спеціалізованих бібліотек. На цій системі зоснована система Бібліотеки Конгресу. Вперше опублікована в 1876 році, вона постійно оновлюється, включаючи нову інформацію доступу необхідну як для традиційних бібліотек, так і для електронних середовищ.

*Універсальна десяткова класифікація* (УДК, UDC – Universal Decimal Classification) - міжнародна схема загальної класифікації [6], що прагне покрити всі області знання. Її джерела йдуть до DDC, що адаптована наприкінці 19 сторіччя Paul Otlet й Henri LaFontaine при спробі створити універсальну бібліографію. Донедавна відповідальність за схему лежала на FID (Federation Internationale de Documentation), потім в 1992 р. вона була передана видавничому консорціуму (UDC Consortium). Первісні цілі використання для впорядкування й індексування вмісту бібліографій, що згодом знайшло своє застосування й для індексування й пошуку в комп'ютерних системах. Схема складається з 60,000 класів (розділів і підрозділів), а також набору додаткових таблиць опису країн, областей і т.п.

УДК і DDC мають ієрархічну структуру, що забезпечує систематизацію (систематичне впорядкування) предметів. Обидві схеми не мають строго впорядковану структуру, використовуючи символи (допоміжні таблиці) для того, щоб дати можливість зібрати складені предмети. Існує багато подоби між позначеннями цих двох схем (УДК була спочатку адаптована зі схеми DDC).

10 класів верхнього рівня:

- 0. Загальна частина.
- 1. Філософія. Психологія.
- 2. Релігія. Теологія.
- 3. Суспільні науки.
- 4. (перебуває в розробці).
- 5. Математика й природознавство.
- 6. Прикладні науки. Медицина. Технічні науки.
- 7. Мистецтво. Хобі. Розваги. Спорт.
- 8. Мови. Лінгвістики. Література.
- 9. Географія. Біографія. Історія.

*Система Класифікації й предметні заголовки Бібліотеки Конгресу* (LCC - Library of Congress Classification, LCSH - Library of Congress Subject Headings ) використовується для систематизації книг академічних й університетських бібліотек у США й світі. І є однією з найпоширеніших і старих (більше 100 років) схем класифікації у світі [6].

*Бібліотечно-бібліографічна класифікація* (ББК) – визнана закордонними експертами Національна класифікаційна система Росії. ББК – наймолодша з універсальних класифікаційних систем і тому добре відбиває логіку розвитку й структуру сучасної науки [7]. За своєю потужністю (обсягів понять) вона порівнянна з УДК. По комбінаційних властивостях значно випереджає LCC.

## **2. Представлення**

Будь-яка бібліотека складається з окремих дискретних частин або ресурсів, що називаються «документами» або «об'єктами». Об'єкти цифрової бібліотеки слід належним чином організувати. Метаданні є головним організуючим чинником бібліотеки і містять у собі інформацію про об'єкти. Стандартні метаданні як правило головним чином забезпечують адміністративну інформацію, а також деяку інформацію стосовно змісту об'єктів (наприклад, назву, опис, предмет). Предметна класифікація – будь-яка форма класифікації змісту, що групує об'єкти по предметах, про які в них розповідається.

Зв'язок між предметною класифікацією та метаданими полягає в тому, що метадані (або поля), що безпосередньо описують те, про що йдеться у описуємих об'єктах (наприклад, ключові слова), в переліку дискретних предметів використовують предметну класифікацію. Метадані описують об'єкти і один із способів їх прояву – як вони з'єднують об'єкти по предметах, про які у них іде мова.

Предметна класифікація може мати різні форми в залежності від парадигми застосування: контрольовані словники, ієрархії концептів, таксономії, тезауруси і т.д. Цей широкий спектр структур представлення знань

про ту чи іншу предметну область згідно сучасних поглядів задовольняють терміну «онтологія» і розглядаються як онтології з різною ступінню формалізації.

Предметі класифікації ACM та UDK – відносяться до так званих легковагих онтологій і становлять собою ієрархії концептів або онтологій, що як правило мають лише концепти з відношеннями *IS-A*.

Інформаційна інтеграція в Web залучає безліч архітектурних блоків, які є центром роботи W3C і пов'язаного семантичне співтовариства Web [8]. Ця робота включає механізми кодування й маніпуляції інформації (наприклад, XML [9], RDF [10], XSLT [11]), а також механізми побудови онтологій й міркування (наприклад, RDFS [12], DAML+OIL [13], OWL [14]).

Мови відрізняються своєю термінологією та експресивністю, але онтології, які вони моделюють як правило мають спільні елементи (концепти, атрибути, відношення). Огляд мов представлень онтологій наведено у [15]. Наведемо лише приклади представлення фрагменту онтології ACM у трьох різних форматах (XML, RDFS, OWL). Оскільки сімейство стандартів W3C на представлення онтологій базується на мові XML (Extensible Markup Language), за допомогою якої можна легко та досить наглядно представити ієрархію концептів класифікації CCS ACM.

Представлення фрагменту онтології ACM у форматі XML.

```
<?xml version="1.0" encoding="windows-1251"?>
<node id="acmccs98_ua" label="ACMCCS98_UA">
  <isComposedBy>
    <node id="H." label="Інформаційні Системи">
      <isComposedBy>
        <node id="H.5" label="Інформаційні інтерфейси й представлення">
          <isComposedBy>
            <node id="H.5.1" label="Мультимедійні інформаційні системи">
              <isComposedBy>
                <node label="Анімація"/>
                <node label="Штучні, збільшені й віртуальна реальності"/>
                <node label="Аудіо ввід-вивід"/>
                <node label="Оцінка/методологія"/>
                <node label="Відео"/>
              </isComposedBy>
            </node>
          </isComposedBy>
        </node>
      </isComposedBy>
    </node>
  </isComposedBy>
</node>
```

Однак, така форма представлення як правило не дозволяє підтримку операцій поза його контекстом.

Важливим з практичної точки зору є стандарт RDF (Resource Description Framework) [10], розроблений W3C для створення метаданих, що описують ресурси Web. Він дає можливість специфікації семантики даних, заснованих на XML, стандартизованим, інтероперабельним способом. Модель даних RDF складається з трьох типів об'єктів: *ресурсів* – об'єкти, що посилаються на адресу в Web; *властивостей*, що визначають певні аспекти, характеристики, атрибути чи відношення використовувані для опису ресурсу; *інструкції*, що призначають значення для властивості у певному ресурсі.

Модель даних RDF не забезпечена механізмами для визначення відношень між властивостями (атрибутами) і ресурсами. Це – роль RDF Schema (RDF Schema Specification language) [12], декларативної мови, що використовується для визначення RDF схем. Вона заснована на деяких ідеях представлення знання (семантичних мережах, фреймах і логіці предикатів), але набагато більш проста в реалізації (але й менш виразна) ніж повні мови числень предикатів. Основні класи – клас, ресурс і властивість, крім того можуть бути визначені ієрархії й обмеження типів. Деякі базові обмеження також визначені. Але все ж для онтологічних визначень у RDF Schema не вистачає функцій і аксіом, але можуть бути легко визначені концепти, відношення й екземпляри.

Представлення фрагменту онтології ACM у форматі RDFS:

```
<?xml version='1.0' encoding='UTF-8'?>
<!DOCTYPE rdf:RDF [
  <!ENTITY rdf 'http://www.w3.org/1999/02/22-rdf-syntax-ns#'>
```

```

    <!ENTITY rdf_ 'http://protege.stanford.edu/rdf'>
    <!ENTITY rdfs 'http://www.w3.org/2000/01/rdf-schema#']>
<rdf:RDF xmlns:rdf="&rdf;"
    xmlns:rdf_="&rdf;"
    xmlns:rdfs="&rdfs;">
<rdfs:Class rdf:about="&rdf_;Н.5._Інформаційні_інтерфейси_й_представлення"
    rdfs:label="Н.5. Інформаційні інтерфейси й представлення">
    <rdfs:subClassOf rdf:resource="&rdfs;Resource"/>
</rdfs:Class>
<rdfs:Class rdf:about="&rdf_;Н.5.1._Мультимедійні_інформаційні_системи"
    rdfs:label="Н.5.1. Мультимедійні інформаційні системи">
    <rdfs:subClassOf
rdf:resource="&rdf_;Н.5._Інформаційні_інтерфейси_й_представлення"/>
</rdfs:Class>
<rdfs:Class rdf:about="&rdf_;Анімації"
    rdfs:label="Анімації">
    <rdfs:subClassOf
rdf:resource="&rdf_;Н.5.1._Мультимедійні_інформаційні_системи"/>
</rdfs:Class>
</rdf:RDF>

```

OWL розширює RDF і RDF Schema, забезпечуючи поряд з формальною семантикою додатковий словник, що закладено в основу мови DAML + OIL, а також убудовану підтримку відображення онтології. OWL має три діалекти: OWL Lite, OWL DL і OWL Full. Вони відрізняються складністю і можуть бути використані в різних прикладних програмах у залежності від необхідності або простоти виводу, або формальності описів.

Представлення фрагменту онтології ACM у форматі OWL:

```

<?xml version="1.0"?>
<!DOCTYPE rdf:RDF [<!ENTITY xsd "http://www.w3.org/2001/XMLSchema#">]>
<rdf:RDF
    xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
    xmlns:xsd="http://www.w3.org/2001/XMLSchema#"
    xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
    xmlns:owl="http://www.w3.org/2002/07/owl#"
    xmlns="http://www.owl-ontologies.com/unnamed.owl#"
    xml:base="http://www.owl-ontologies.com/unnamed.owl">
<owl:Ontology rdf:about="">
<owl:Class rdf:about="#Н._Інформаційні_системи">
    <rdfs:label rdf:datatype="&xsd:string"
    > Н. Інформаційні системи</rdfs:label>
</owl:Class>
<owl:Class rdf:about="#Н.5.1._Мультимедійні_інформаційні_системи">
    <rdfs:subClassOf>
    <owl:Class rdf:about="#Н.5._Інформаційні_інтерфейси_й_представлення"/>
</rdfs:subClassOf>
<rdfs:label rdf:datatype="&xsd:string"
    >Н.5.1. Мультимедійні інформаційні системи</rdfs:label>
</owl:Class>
<owl:Class rdf:about="#Н.5._Інформаційні_інтерфейси_й_представлення">
    <rdfs:label rdf:datatype="&xsd:string"
    >Н.5. Інформаційні інтерфейси й представлення</rdfs:label>
</owl:Class>
<owl:Class rdf:ID="Анімації">
    <rdfs:subClassOf>
    <owl:Class rdf:about="#Н.5.1._Мультимедійні_інформаційні_системи"/>
</rdfs:subClassOf>
</owl:Class>
</rdf:RDF>

```

Protégé на сьогодні є найпопулярнішим відкритим середовищем розробки онтологій, що може використовуватися для завантаження, редагування та зберігання онологій у різних форматах, а саме

стандартного текстового, баз даних, UML, мов XML, XOL, SHOE, RDF і RDFS, DAML+OIL, OWL. Редактор онтологій Protégé дозволяє проектувати онтології розвертаючи ієрархічну структуру абстрактних або конкретних класів або слотів. На основі сформованої онтології Protégé може генерувати форми одержання знань для введення екземплярів класів та підкласів. Може розширюватись за допомогою плагінів:

- графічної візуалізації онтологій (Jambalaya, OntoViz, OWLViz);
- об'єднання й групування онтологій (PROMPT);
- керування великими онлайнними джерелами знань (UMLS і WordNet), доступ до ОКВС онтологій (ОКВС);
- побудови та виконання обмежень (PAL);
- машини висновку, що використовують Jess, Prolog, FLogic, FaCT, Algernon (відповідно Jess, Prolog, FLORA, OIL та Algernon).

OWL Plugin є плагіном Protégé для підтримки OWL. Він може використовуватися для завантаження та збереження OWL-файлів у різних форматах, редагування онтології на OWL за допомогою зручної графічної форми, та виконання інтелектуального міркування на основі дескриптивних логік (DL). Рис. 1 дає приклад візуалізації онтології ACM у форматі OWL за допомогою графічного візуалізатора OWLViz.

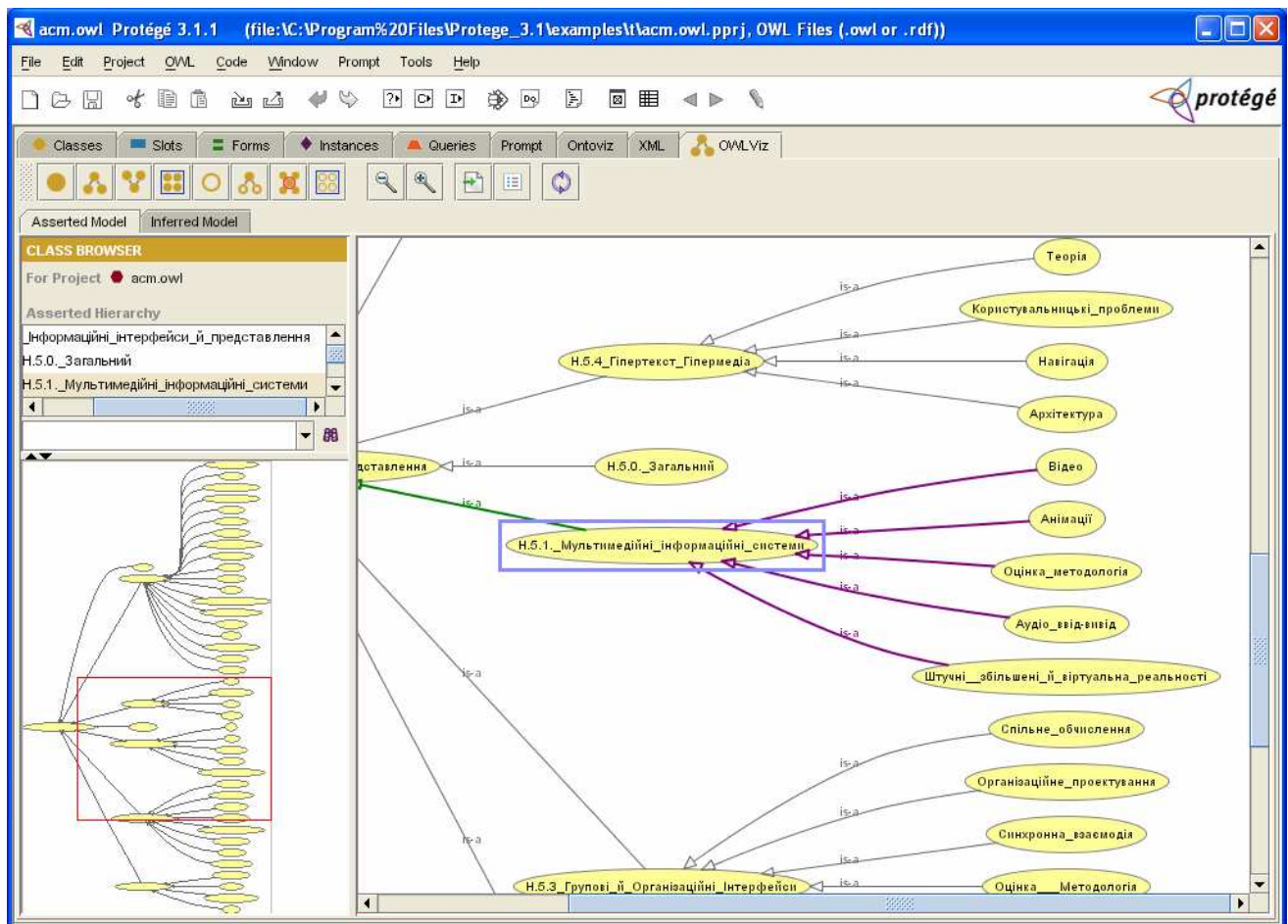


Рис. 1. Візуалізація онтології ACM за допомогою OWLViz

### 3. Інструменти для відображення, вирівнювання та об'єднання онтологій

Відображення онтологій це серйозна проблема. Вона вийшла на передній план за останні роки, зокрема в сфері цифрових бібліотек, у зв'язку з їх намаганням об'єднати гетерогенні дані, тому часто розглядається як частина більш вагомій задачі вирівнювання та об'єднання онтологій.

Інструменти об'єднання онтологій допомагають користувачам знайти подібність і відмінність між вхідними онтологіями й створюють результуючу онтологію, що містить елементи вхідних онтологій. Для досягнення цієї мети вони автоматично визначають відповідності між концептами у вхідних онтологіях або забезпечують середовище, де користувач може легко знайти й визначити ці відповідності. Ці інструменти відомі як інструменти відображення, вирівнювання й об'єднання онтологій, тому що вони виконують подібні операції для процесів відображення, вирівнювання й об'єднання.

Відображення (mapping) онтологій полягає в знаходженні семантичних зв'язків (або відношень) подібних елементів з різних онтологій. Відношення може бути еквівалентністю, узагальненням/спеціалізацією та ін.



Вирівнювання (alignment) онтологій полягає у тому, щоб установити різні види відповідності (або зв'язку) між двома онтологіями, а потім повторно зберегти вхідні онтології й таким чином, надалі використовувати інформацію один одного.

Об'єднання (merging) онтологій – генерація однієї погодженої онтології із двох вхідних, причому вхідні онтології залишаються незмінними [16–17].

Інструментальні засоби, які мають справу із знаходженням відповідності між онтологіями, можна класифікувати наступним чином [18]:

- для об'єднання двох онтологій з метою створення однієї нової (PROMPT, Chimaera, OntoMerge);
- для визначення функції перетворення з однієї онтології в іншу (OntoMorph);
- для визначення відображення між концептами у двох онтологіях, знаходячи пари відповідних концептів (наприклад, GLUE, FCA-Merge);
- для визначення правил відображення для зв'язку тільки релевантних частин вихідних онтологій (ONION).

Коротко розглянемо вищезгадані засоби.

PROMPT [18] – доповнення до системи Protege, реалізоване у вигляді плагіна, служить для об'єднання й групування онтологій. При об'єднанні двох отологій (рис. 2) PROMPT створює список запропонованих операцій. Операція може складатися, наприклад, з об'єднання двох термінів або копіювання термінів у нову онтологію. Користувач може виконати операцію, вибираючи одну із запропонованих або визначаючи безпосередньо операцію. PROMPT виконує обрану операцію й додаткові зміни, викликані цією операцією. Потім список запропонованих операцій модифікується й створюється список конфліктів і можливих рішень цих конфліктів. Це повторюється доти, доки не буде готова нова онтологія.

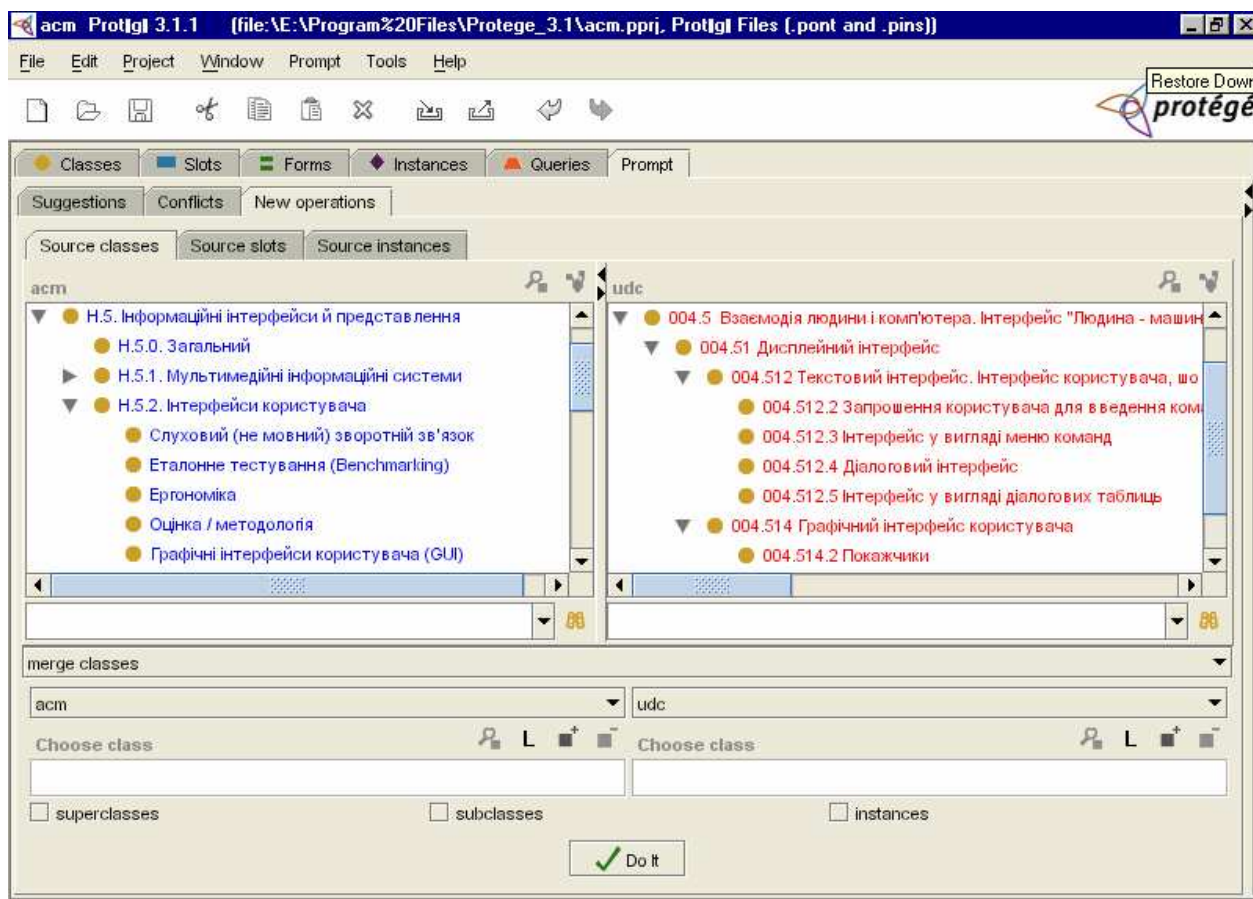


Рис. 2. Приклад злиття онтологій ACM та UDC за допомогою Prompt

Chimaera [19] – інтерактивний інструмент для об'єднання, заснований на редакторі онтологій Ontolingua. Chimaera дозволяє користувачеві поєднувати онтології, розроблені в різних формалізмах. Єдине таксономічне відношення, що розглядає Chimaera – відношення підклас – суперклас. Chimaera найближчий до PROMPT. Однак оскільки він використовує у своєму аналізі тільки ієрархію класу, він пропускає багато з відповідностей, які знаходить PROMPT.

OntoMerge [20] використовує універсальну машину висновку на дозвіл на трансляцію двох онтологій, що відображаються одна на одну. В OntoMerge відповідності між онтологіями виражаються набором *аксіом сполучення*, що зв'язують класи та властивості двох вхідних онтологій. Потім дві вхідні онтології та набір

аксіом сполучення розглядаються як окрема теорія засобом доведення теорем, оптимізованого для задачі трансляції онтології.

OntoMorph [21] визначає набір операторів перетворення, які можна застосувати до онтології. Потім людина-експерт використовує початковий список пар і вхідних онтологій для визначення набору операторів, які повинні застосуватися до вхідних онтологій для усунення розходжень між ними, і OntoMorph застосовує ці оператори. Таким чином, сукупність операцій може виконуватися за один крок. Однак, людина-експерт не одержує ніякого керівництва за винятком початкового списку пар.

FCA-Merge [23] – метод для порівняння онтологій, які мають набір загальних екземплярів або набір загальних документів, що анотуються за допомогою концептів вхідних онтологій. Грунтуючись на цій інформації, FCA-Merge використовує математичні методи з Formal Concept Analysis для того щоб зробити решітку концептів, що зв'язує концепти вихідних онтологій. Алгоритм пропонує відношення еквівалентності й підклас-суперклас. Потім інженер онтології може аналізувати результат і використовувати його як керівництво для створення об'єднаної онтології.

Система ONION (ONtology compositIION) [24] заснована на алгебрі онтології. Тому, вона надає інструменти для визначення правил артикуляції (з'єднання) між онтологіями. Правила артикуляції звичайно враховують тільки релевантні частини вхідних онтологій. Для того щоб запропонувати з'єднання, ONION використовує і лексичні методи, і методи на основі графів. Метод знаходження лексичної подоби між іменами концептів використовує словники й методи семантичної індексації, засновані на місцезнаходженні групи слів у тексті.

Створення відображень трудомісткий та піддатний до помилок процес навіть для людини. Багато сучасних інструментів відображення є напівавтоматичними, допомагаючи користувачеві в інтерактивному режимі. Деякі відображення виконуються повністю автоматично, використовуючи поєднання методів лінгвістики та машинного навчання GLUE [25]. Однак результати ще більш піддатні до помилок та онтології що використовуються цими методиками неформальні. Слід вибрати такий інструмент відображення, де можна отримати найбільш економно за трудозатрами автоматизацію при забезпеченні сприятливого рівня точності. Проблема покращити точність повністю автоматизованих методів, доповнюючи їх інтерактивними підходами з залученням людини.

#### 4. Приклади відображення онтологій

Відображення повинно точно визначати семантичне відношення між двома сутностями, що належать різним онтологіям. Відношення може бути еквівалентністю так само як спеціалізацією/узагальненням або будь-яким іншим видом відношень [26].

Процес відображення буде розглянуто на прикладі підкласів „004.5 Взаємодія людини і комп'ютера. Інтерфейс "Людина - машина". Інтерфейс користувача. Середовище користувача” класифікації UDC та „Н.5 Інформаційні інтерфейси й представлення” класифікації АСМ (рис. 3, 4).

В нашому випадку для побудови відображення використовувалися такі відношення: еквівалентність; підклас; суперклас.

У таблиці наведені зв'язки між фрагментами онтологій АСМ та UDC, що знайдені у процесі відображення:

Таблиця

АСМ	Відношення	UDC
Н.5.2 Стили взаємодії (наприклад, команди, меню, форми, пряма маніпуляція)	Суперклас	004.512.2 Запрошення користувача для введення команди (Командний "промт") ∪ 004.512.3 Інтерфейс у вигляді меню команд ∪ 004.512.4 Діалоговий інтерфейс ∪ 004.512.5 Інтерфейс у вигляді діалогових таблиць ∪ 004.514.2 Показчики ∪ 004.514.4 Меню, що випадає або впливає
АСМ	Відношення	UDC
Н.5.2 Графічні інтерфейси користувача (GUI)	Еквівалентно	004.514 Графічний інтерфейс користувача
Н.5.2 Системи керування вікнами	Еквівалентно	004.514.6 Віконне середовище Вікноподібний інтерфейс
Н.5.2 Природна мова ∪ Н.5.2 Голосовий ввід-вивід	Підклас	004.522 Мовний інтерфейс

Н.5.2. Слуховий (не мовний) зворотній зв'язок	Підклас	004.523 Використання інших звуків, крім мови
Н.5.4. Гіпертекст/Гіпермедіа	Еквівалентно	004.55 Гіпермедіа. Гіпертекст
Н.5.2. Навчання, довідка, і документація	Суперклас	004.58 Допомога користувачу

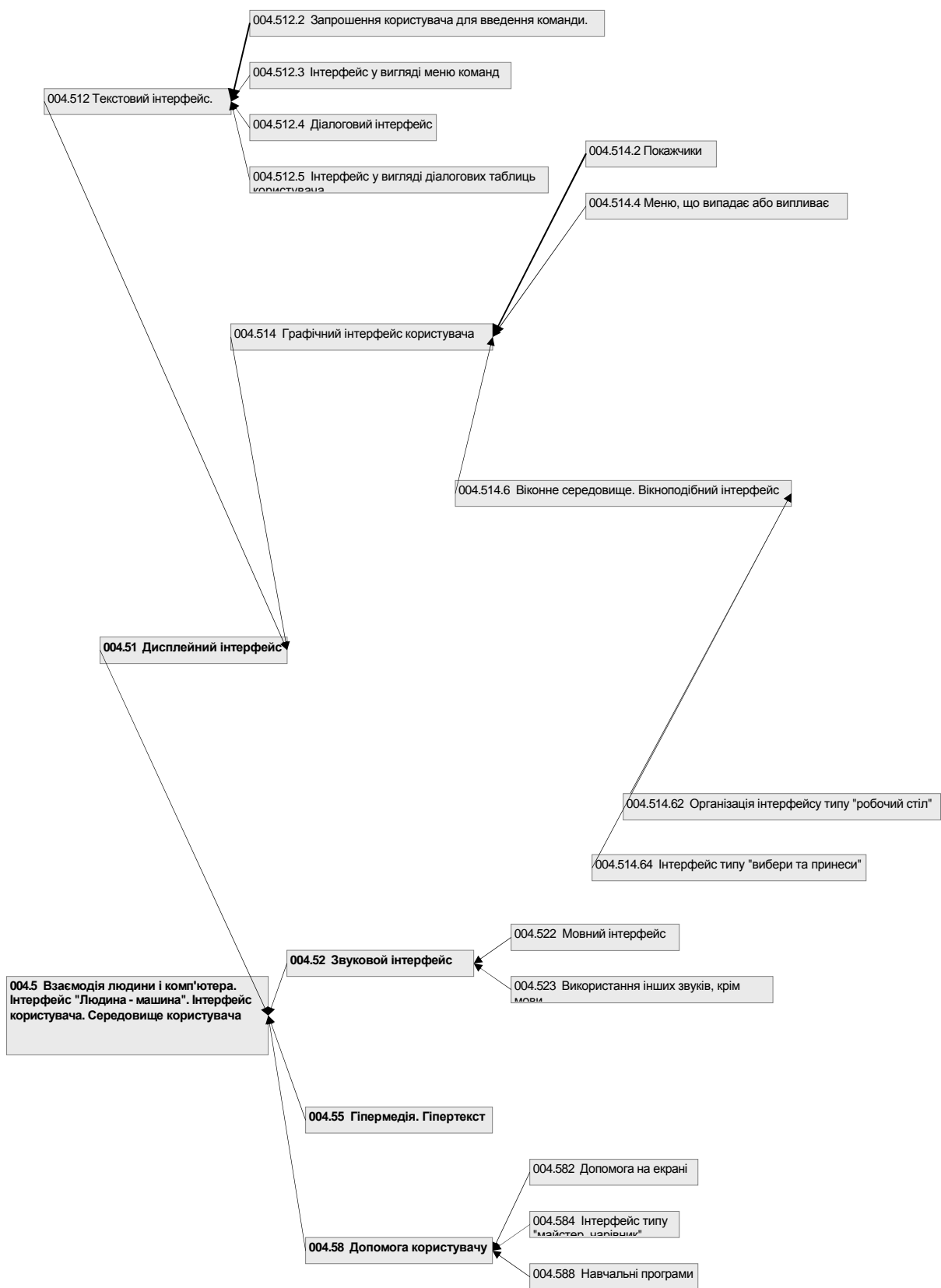


Рис. 3. Фрагмент предметної класифікації UDC



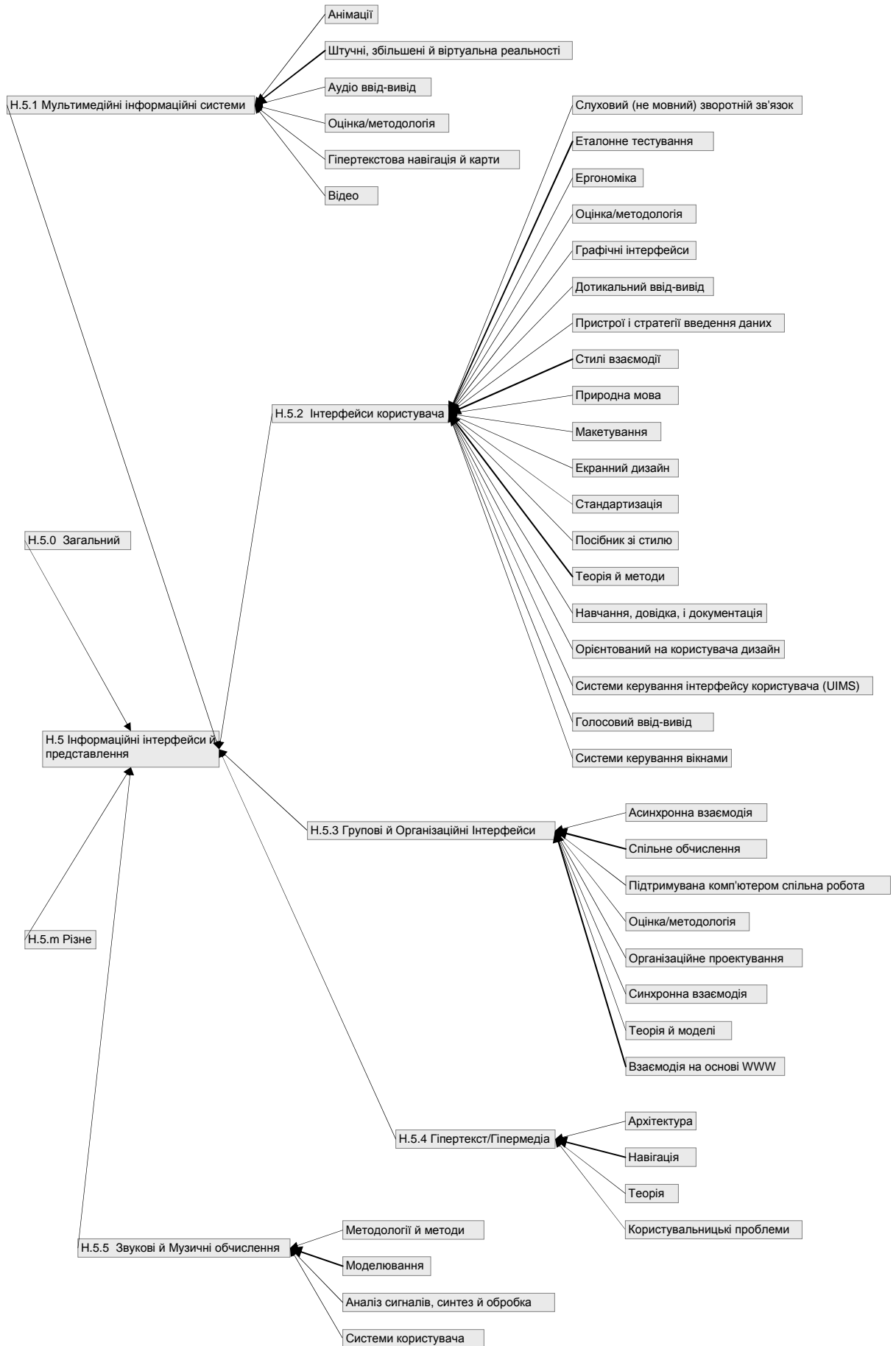


Рис. 4. Фрагмент предметної класифікації ACM

На рис. 5 показано графічне представлення відображення.

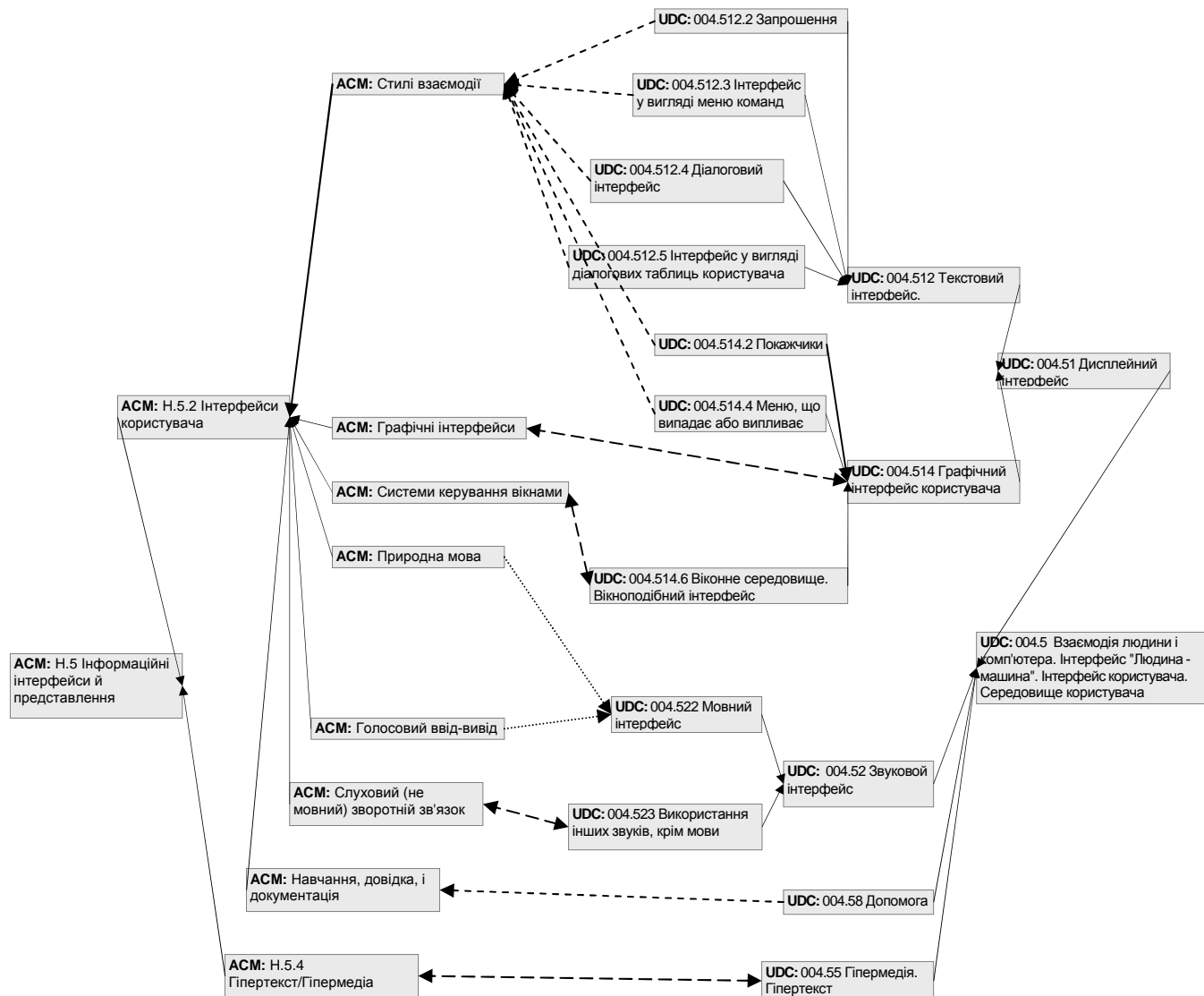


Рис. 5. Графічне представлення відображення

Ще один спосіб виразити інформацію відображення – використати вирази самої мови онтології для того щоб виразити взаємозв'язки. Вище свідчилось про те, що мова OWL реалізує можливості для відображення онтологій. Тому таким чином можна представляти не тільки самі онтології, але і їх відображення. На представленому фрагменті OWL-коду, показано спосіб за допомогою якого можна визначити клас в одній класифікації, в даному фрагменті (&ast; Н.5.2.\_Інтерфейси\_користувача), а потім об'явити його еквівалентним існуючому класу (&uac1d; 004.514\_Графічний\_інтерфейс\_користувача). Як бачимо, для здійснення такого відображення у OWL є властивості owl:equivalentClass (класи мають однакові екземпляри) та owl:equivalentProperty. В OWL Full між двома класами можна використовувати owl:sameAs, щоб показати їх ідентичність у всіх відношеннях.

```
<owl:Class rdf:ID="Графічні_інтерфейси_користувача_GUI">
  <rdfs:subClassOf>
    <owl:Class rdf:about="#Н.5.2._Інтерфейси_користувача"/>
  </rdfs:subClassOf>
  <owl:equivalentClass
rdf:resource="&uac1d;004.514_Графічний_інтерфейс_користувача">
  </owl:Class>
```

## 5. Висновки

Предметні каталоги, тезауруси використовуються у якості структуруючого елемента у цифрових бібліотеках. Через відсутність єдиних стандартів таких каталогів та у зв'язку з їх постійною зміною для

забезпечення інтероперабельності виникає проблема їх відображення, з'єднання, вирівнювання (автоматичного або напівавтоматичного).

Як було виявлено на прикладі відображення, з'єднання, вирівнювання предметних каталогів потребує значних інтелектуальних зусиль. Тому опис термінів та відношень предметних класифікацій потребує застосування більш виразніших мов (представлень) з одного боку та інструментарію що міг би працювати з такими представленнями з іншого. Ясне онтологічне погодження всіх понять, де текстовий опис та формальне визначення понять та їх відношень в обох каталогах було б надзвичайно корисно.

Ми розцінюємо одержаний методологічний досвід на розглянутому прикладі відображення фрагментів предметних класифікацій АСМ та УДК з застосуванням онтологічних інструментів як корисний проміжний результат для майбутніх досліджень.

1. *Резиченко В.А., Проскудина Г.Ю., Овдій О.М.* Создание цифровой библиотеки коллекций периодических изданий на основе Greenstone. Электронные библиотеки. 2005. — 8. - Вып. 6 — <http://www.elbib.ru/index.phtml?page=elbib/rus/journal/2005/part6>.
2. *Овдій О.М., Проскудина Г.Ю.* Обзор инструментов инженерии онтологий. Электронные библиотеки. 2004. — 7. Вып. 4 — <http://www.elbib.ru/index.phtml?page=elbib/rus/journal/2004/part4>.
3. *ACM Computing Classification System (CCS)* - <http://www.acm.org/class/1998>.
4. *How to classify works using acm's computing classification system*, [http://www.acm.org/class/how\\_to\\_use.html](http://www.acm.org/class/how_to_use.html).
5. *Computing Classification System 1998: Current Status and Future Maintenance*. Report of the CCS Update Committee, Neal Coulter (Chair), James French, Ephraim Glinert, Thomas Horton, Nancy Mead, Roy Rada... — <http://www.acm.org/class/1998/ccsup.pdf>.
6. *Resource Description and Classification*. Being a collection of references on matters of Subject Classification, Taxonomies, Ontologies, Indexing, Metadata, Metadata Registries, Controlled Vocabularies, Terminology, Thesauri, Business Semantics — <http://xml.coverpages.org/classification.html>.
7. *БИБЛИОТЕЧНО-БИБЛИОГРАФИЧЕСКАЯ КЛАССИФИКАЦИЯ* — [http://www.rsl.ru/tot.asp?7\\_8\\_1.htm#1](http://www.rsl.ru/tot.asp?7_8_1.htm#1)
8. *Berners-Lee T., Hendler J., Lassila O.* "The Semantic Web," *Scientific American*(50), May, 2001.
9. <http://www.w3.org/XML/>.
10. <http://www.w3.org/TR/PR-rdf-syntax/>.
11. <http://www.w3.org/TR/xslt>.
12. <http://www.w3.org/TR/rdf-schema>.
13. *DAML+OIL (March 2001) Description*. <http://www.w3.org/TR/daml+oil-reference>.
14. <http://www.w3.org/TR/owl-features/>.
15. *Овдій О.М., Проскудина Г.Ю.* Онтології у контексті інтеграції інформації: представлення, методи та інструменти побудови // Проблеми програмування. Спеціальний випуск, 2004. — № 2, 3.— С. 353–365.
16. *Tessaris: D2.2.1: Specification of a common framework for characterizing alignment* Technical report. *P. Bouquet, J. Euzenat, E. Franconi, L. Serafini, G. Stamou, S.* // NoE Knowledge Web project deliverable. — 2004.
17. *Noy N., Musen M.* SMART: Automated Support for Ontology Merging and Alignment // Stanford Medical Informatics, Stanford Univ. — 1999.
18. *Noy N., Musen M.* The PROMPT Suite: Interactive Tools For Ontology Merging And Mapping // Stanford Medical Informatics, Stanford Univ. — 2003.
19. *McGuinness D., Fikes R., Rice J., Wilder S.* An environment for merging and testing large ontologies // In Proc. of the Seventh Int. Conf., KR2000, Morgan Kaufmann Publishers, San Francisco, CA, 2000.
20. *Dou D., McDermott D., Qi Peisen.* Ontology translation on the Semantic Web //In Int.. Conf. on Ontology, Databases and Applications of Semantics, 2003. — [http://www.cs.yale.edu/~dvm/daml/ontomerge\\_odbase.pdf](http://www.cs.yale.edu/~dvm/daml/ontomerge_odbase.pdf).
21. *Chalupsky H* OntoMorph: A translation system for symbolic knowledge // In: Cohn A. G., Giunchiglia F., Selman B. (Eds.), Principles of Knowledge Representation and Reasoning // Proc. of the Seventh Intern. Conf. (KR2000). Morgan Kaufmann Publishers, San Francisco, CA. — 2000.
22. *Mena E., Illarramendi A., Kashyap V., Sheth A.* OBSERVER: An approach for query processing in global information systems based on interoperation across preexisting ontologies // Distributed and Parallel Databases, An International J. — 2000. — 8. N.2.
23. *Stumme G., Medche A.* FCA-Merge: Bottom-up merging of ontologies // 7th Int. Conf. on Artificial Intelligence, (IJCAI'01), Seattle, WA, 2001. — P. 225–230.
24. *Mitra P., Wiederhold G., Decker S.* A scalable framework for interoperation of information sources //The 1st Intern. Semantic Web Working Symp., SWWS'01, Stanford University, CA, 2001.
25. *Doan A., Madhavan J., Domingos P., Halevy A.* Learning to map between ontologies on the Semantic Web // The Eleventh Intern. WWW Conference. Hawaii, US. — 2002.
26. *Doerr M., Hunter J., Lagoze C.* Towards a Core Ontology for Information Integration // submitted to JoDI, October 2002.