

УДК 681.3.

*С.Л. Полянничко*

## **КЕРУВАННЯ СТРАТЕГІЯМИ ЗМІНИ РЕЛЕВАНТНОСТІ ПРИ ПОШУКУ ПРОГРАМНИХ КОМПОНЕНТ В РЕПОЗИТОРІЇ ОНТОЛОГІЧНОЇ СИСТЕМИ**

Тема дослідження даної роботи - багатоаспектна атестація та пошук програмних компонент в онтологічній системі. Пропонується метод використання онтологій аспектів для автоматизованого керування зміною стратегії встановлення релевантності при пошуку програмних компонент в онтологічній системі.

Одна з базових проблем у побудові інформаційно-пошукових систем (ІПС) - проблема визначення критеріїв релевантності образів ресурсів у процесі інформаційного пошуку. Користувач формулює свою інформаційну потребу у вигляді пошукового виразу, який прийнято називати пошуковим запитом (ПоЗ), а система виконує запит шляхом співставлення відповідного пошукового виразу зі специфікаціями ресурсів в ІПС, які є образами ресурсів (ПоБ). Форма представлення такого ПоЗ зафіксована для конкретної ІПС.

При первинному формулюванні запиту пошук ресурсів може дати результати, які не відповідають потребам користувача ІПС. У випадку занадто конкретного ПоЗ системою не буде знайдено жодного ПоБ ресурсу, який відповідає ПоЗ. Іншим граничним випадком є такий, коли у відповідь на узагальнений запит користувача система вкаже йому на таку множину релевантних ПоЗ ресурсів, потужність якої не дозволяє здійснити якісний відбір ресурсів. У обох випадках виникає потреба виробити ефективні стратегії зміни первинного пошукового виразу шляхом встановлення спеціального критерію релевантності між ПоЗ та ПоБ.

Актуальна задача сучасних ІПС - побудова бази знань щодо ресурсів на основі онтологічних моделей предметних областей [1]. Переважний напрямок таких досліджень - застосування онтологій як засобу керування при впровадженні різних послуг в ІПС [2]. Онтологія, або понятійна база, визначає не тільки узгоджену, уніфіковану термінологію, якою повинні корис-

туватися різні носії інтересів та користувачі ІПС, але і відносини між поняттями, фіксує їх інтерпретацію. У нашому випадку під інтерпретацією понять або відношень будемо розуміти їх неформальні визначення або коментарі.

Відмінна риса онтологій аспектів - їх використання не тільки для навігації у просторі загальних знань щодо домену предметної області, скільки для формування знань щодо конкретної програмної компоненти в різних аспектах. Репозиторії у складі онтологічної системи [3] націлені на зберігання образів готових програмних компонент та подальше їх використання у нових розробках.

У даній роботі пропонується метод використання предметних онтологій для автоматизованого керування стратегіями зміни релевантності за допомогою апарату критеріїв релевантності при пошуку в репозиторії.

У даній онтологічній системі аспекти представлення знань щодо програмних компонент будуються у відповідності з методологією, яка набуває статусу стандарту для представлення артефактів програмної інженерії, а саме методологією UML [4]. Такі онтологічні аспекти містять відомості про всі сценарії (роботи), притаманні домену, про класи об'єктів, що діють у сценаріях, визначені для об'єктів функції (методи), стани та атрибути тощо. Таким чином усі концепти предметних знань, що використовуються, підпорядковуються концептам UML за допомогою передбачених для них в UML відношень та стереотипів.

У проєкті, що розглядається в даній роботі, запропоновано інтегрований програмний комплекс, який дозволяє накопичити онтології аспектів для опису різних програмних компонент в різних предметних галузях. Цей програмний комплекс містить:

- графічний редактор онтологій, призначений для первинної побудови та подальшої модифікації онтологій аспектів в рамках окремого домену;
- конструктор запитів та образів, призначений для формування ПоЗ та ПоБ компоненти на базі відповідним онтологічним аспектам;
- репозиторій образів компонент;
- машина пошуку образу компоненти в репозиторії образів згідно поданому ПоЗ;
- сервісні функції, які дозволяють варіювати змінами стратегії релевантності при пошуку за допомогою наперед визначеного набору критеріїв релевантності між ПоЗ та ПоБ;
- система звітності про результати пошуку в репозиторії.

Зупинимось на особливостях представлення ПоБ на базі багатьох онтологічних аспектах та на апараті критерію релевантності, який дозволяє впровадити ефективний пошук у багатоаспектній онтологічній системі.

### **1. Моделювання ПоБ програмної компоненти та запиту на її пошук в багатоаспектній онтологічній системі**

Складність сучасних компонент програмної інженерії вимагає спеціальних підходів для формального опису їх можливості та певних прийомів для їх інтеграції у нову розробку. Одним з таких прийомів, на наш погляд досить ефективним, є декомпозиція розгляду суцільної проблеми на окремі аспекти, тобто з позицій окремих точок зору або окремих носіїв інтересів. Прикладами різних аспектів, широко застосовуваних при розгляді компонент, є їх функціональні властивості, виконавчі показники, паспортні дані, показники якості,

захищеності, візуальні чи аудіо засоби представлення даних. Аспектне представлення дозволяє побудувати для кожного аспекту свою систему навігації при інформаційному пошуку. Аспектний підхід представлення знань в галузі програмної інженерії набув широкого вживання як розвиток фасетного підходу. При фасетному підході опису ресурсів кожному виділеному аспекту буде відповідати своя фасета як частина бази знань домену.

У залежності від предметних областей, окремі з названих аспектів можуть, в свою чергу, поділятися на під-аспекти.

Аспектний склад онтологічної системи для представлення знань щодо програмних компонент наперед визначений для стандартизованого опису кожної ПК в домені. Онтологічна модель аспектів дозволяє проводити нормалізацію лексики щодо представлених знань, атестацію та пошук компонент в системі. В даній роботі представляється модель онтологій аспектів, які побудовані за допомогою концептуальних графів. Такий аспектний опис компонент дозволяє побудувати шаблон, за допомогою якого атестуються компоненти та зберігаються образи програмних компонент. Графічний образ компоненти за допомогою спеціального алгоритму транспонується у векторне текстове представлення. У подальшому текстовий онтологічний багатоаспектний образ програмної компоненти транспонується в опис на HTML подібній мові та зберігається в спеціальному репозиторії. Такий опис компоненти фіксується за допомогою спеціальних тегів, кожний з яких відповідає аспектному розділу опису компоненти в онтологічній системі. Вміст окремого тегу- текстовий образ компоненти, який побудований за допомогою відповідного онтологічного аспекту.

Склад аспектів домену опису ПК поділяється на ті, що представляють технічні властивості та семантичні властивості. Вбудовані стандартні аспекти дозволяють атестувати ПК з точки зору технології інтеграції її в нову розробку (або технології її ревикористання [5]), а саме:

аспект інтерфейсу, аспект розгортання, аспект реалізації.

Ці аспекти є базисом для аспектного опису будь-якої ПК. Вони складають інваріант для різних галузей, в яких можуть бути описані різні ПК. Кожна з таких онтологій - аспектів має певну наперед визначену структуру. До її особливостей можна віднести таку, що родова частина кожного з цих аспектів (ярус метакласифікаторів) є незмінною при підключенні такого аспекту до нового домену. Але змінною частиною є проміжний рівень концептів та нижній рівень концептів – значень, які складають нижні яруси онтологічної ієрархії.

Окрім вбудованих стандартних аспектів до атестації компоненти можуть бути підключені аспекти, які мають різне представлення в різних доменах та які дозволяють представити семантику та індивідуальні особливості кожної компоненти. Це такі: аспект типізації, семантичний аспект, аспект паспортних даних, аспект ролей.

Особливістю цих аспектів (особливо семантичного аспекту) є те, що вони змінюються при підключенні до нового домену повністю (разом зі своєю родовою (скелетною) частиною).

У процесі інженерії домену для атестації програмних компонент для кожного з наведених аспектів були наперед визначені базові родові концепти (які породжують скелетні гілки), основний склад концептів проміжного рівня, типи відношень, які характерні для кожного з цих аспектів та типи варіантності в кожному аспекті. Для прикладу наведемо особливості онтології аспекту ролей.

**Джерела.** Джерелами для атестації компоненти в цьому аспекті можуть бути специфікатори діаграм сценаріїв UML або ER-діаграми.

**Родові концепти для скелетних гілок:**

- ролі;
- сценарії;
- процеси;
- стани;
- вимоги.

**Правила побудови онтології аспекту ролей.** При побудові такого аспекту виходимо з наступних положень:

- кожний стан об'єкта визначається за допомогою значень його атрибутів;
- кожний стан може визначатися своїм зв'язком з класом та з атрибутами цього класу, (або з атрибутами, які наслідуються від іншого класу);
- кожний об'єкт може виконувати роль;
- кожний об'єкт може мати операції;
- стани можуть мати переходи, причому перехід для стану змінює його атрибути;
- стани можуть бути складними та складатися з інших простих станів та переходів зі стану в стан;
- перехід зі стану в стану асоціюється з викликом операції;
- стан може мати суперстан та підстан (причому вони наслідують атрибути).

Аспект ролей має **базові відношення:**

- <виконує> “роль”;
- <є> “процесом”;
- <є> “сценарієм”;
- <має> “ціль”;
- “процес” <має> “підпроцес”;
- <відповідає> “вимогам”;
- <виконує> “дію”;
- “процес” <підтримує> “дію”;
- “дія” <має> “вхід”;
- “дія” <має> “задачу”;
- “дія” <має> “вихід”;
- <виконується на> “класі”;
- <виконується на> “підкласі”;
- “стан” <має> “атрибут стану”;
- “атрибут стану” <має> “значення”;
- “стан” <асоціюється з> “класом”;
- <має> “опрацію”;
- “стан” <має> “перехід”.

**Варіантність** аспекту ролей. Варіантність компонент у кожному аспекті представляється за допомогою галуження онтологічної ієрархії. Типи галуження онтології для відображення варіантності в різних аспектах можуть бути обов'язкові, необов'язкові або альтернативні. Для кожного типу аспекту встановлюється свій тип варіантності [6]. До варіантності аспекту ролей можна віднести наступні:

- процес може змінюватися завдяки різним цілям (варіюється набір цілей);
- кожна з цілей компоненти досягається різними сценаріями (варіюється набір сценаріїв);
- сценарії можуть працювати з різними об'єктами та класами (варіюється набір класів).

Аспект вважається знаним у репозитарії, якщо у його базу знань входить відповідна аспекту онтологія. Вищеперелічені аспекти вважаються обов'язковими для роботи репозитарію. Це означає, що ПОБ, як і ПОЗ складається з фрагментів, кожен з яких може бути атестаційним або пошуковим виразом відносно одного із знаних аспектів, а між наявними фрагментами вважається встановленим відношення диз'юнкції.

У процесі багатоаспектної атестації (анотування) кожний компонент представляється в домені предметної галузі за допомогою наступних знань:

- сукупності аспектів, за якими він атестується в домені;
- родових концептів, які визначають гілки в аспекті, значущі для атестації компонента;
- концептів проміжного рівня, які атестують його за кожним аспектом;
- значень для атестаційних концептів проміжного рівня;
- відношень, між атестаційними концептами;
- тлумачень атестаційних концептів;
- груп синонімів для атестаційних концептів;
- атрибутів приналежності до аспекту в домені;
- коментарів;
- графічній візуалізації образу компонента в окремому аспекті;
- текстового образу компонента;
- HTML документа, який відповідає багатоаспектному текстовому та графічному образу компонента;
- URL посилань до документів, які можуть розширити інформаційне представлення компонента в системі.

Для представлення власне онтологічній аспектів обрано форму концептуальних графів.

Для концептуальних графів проблема встановлення контекстної відповідності зводиться до процесу керованого онтологією пошуку на графових даних. Концепти та відношення вважаються порівнянні, якщо онтологія демонструє, що між ними встановлено зв'язок. Разом з цим використовується словникова підтримка онтології, тобто вузли концептуального графа прив'язуються до відповідних лексичних одиниць домену. При цьому підтримуються відповідні семантичні обмеження.

Окрім суто онтологічних атрибутів (назва концепту, назва відношення, тлумачення концепту) вершині графа відповідає набір графічних атрибутів як розмір, форма, місце та ін. Кожній дузі графа відповідають наступні атрибути – стиль дуги, які визначають стандартні відношення в онтології та мітка дуги, яка відповідає назві нестандартного відношення в онтології домену. Ці графічні атрибути можуть обиратися для створення дружнього інтерфейсу згідно персональним смакам користувача або встановлюватися системою і є засобами візуалізації онтології.

Підсумовуючи вищеописане, можемо вважати наступне:

- багатоаспектна база знань про програмні компоненти представляється “лісом” дерев – онтологій. Такий ліс дерев онтологій будемо вважати онтологічною моделлю домену “можливі компоненти певної предметної області” при чому кожна із онтологій розглядається як концептуальна модель відповідного аспекту ;
- кожне з дерев “лісу” відповідає одному з аспектів представлення знань щодо компонента;
- кожній вершині обов'язково відповідає дескриптор, який має своє тлумачення в словнику домену;
- існує виділена вершина, яка є входом до представленого графом аспекту. Причому кожна вершина графа повинна бути досяжною із цієї вхідної вершини (кореня аспекту).

На фазі анотування – побудови ПОБ ресурсу вершини та дуги онтології обраного аспекту зазначаються (або “активуються”) користувачем за допомогою графічного інтерфейсу, і в подальшому трансформуються в атестаційний розділ ПОБ щодо відповідного ресурсу, представлений у HTML-подібній мові.

Подальшими етапом методики стає алгоритм трансформування багатоаспектного ПоБ компоненти в текстовий опис компоненти на HTML подібній мові опису компоненти. В такому HTML представленні за допомогою HTML маркерів виділяються розділи, які відповідають назвам аспектів. При цьому ім'я розділу є ім'ям аспекту. Підрозділам аспекту відповідають спеціальні теги [7]. Треба зазначити, що HTML структура дозволяє ввести ієрархію включених один в одного тегів, тим самим підтримати ієрархію онтологічних метакласифікаторів. Вміст тегів – ПОБ компоненти в поточному підрозділі аспекту, яке будується за правилами мови онтологічного опису та складається з концептів середнього та нижнього рівнів онтологічної ієрархії та їх значень.

Таким чином активована частина графа онтології трансформується в граф ПоБ, причому для кожного поняття зберігається семантична відповідність концепту онтології у рамках домену. Саме така функція використовується і при формуванні пошукового образу компонента. Перед трансформуванням здійснюється семантична валідація графа ПоБ компонента на базі словника онтології. Надалі ПоБ компоненти запам'ятовується на спеціальній мові опису пошукового образу програмної компоненти МОК, яка детально описана в [2].

Тобто стверджуємо, що інформаційна модель компонента в домені – це його багатоаспектний пошуковий образ, який має семантичну підтримку в онтологічній системі. Багатоаспектний образ компонента відповідно представляється конкатенацією атестаційних розділів по кожному, щонайменше одному, аспекту в онтологічній моделі домену предметної області.

Запити на пошук адекватного вимогам користувача компонента будується на базі трьох елементів. Перший елемент – онтологічні образи, що співставляються (ПоБ та ПоЗ), другий – локалізація (місце співставлення, тобто всатновлення поточного аспекту для проведення співставлення), третій – умови співставлення (предикати співставлення).

У нашому випадку в ролі першого елемента виступає запит (ПОЗ компоненти) – це граф, вершини якого пов'язані функцією з концептами онтології. Побудований за цим графом кортеж співставляється з аналогічно побудованими кортежами в репозиторії ПК. Найдені відповідні кортежі називають образами компонент (ПОБ компоненти). Місце, де проводиться співставлення в онтології, визначається набором аспектів, які фігурують в ПОЗ.

За кожним аспектом в ПОЗ здійснюється операція трансформування графа запиту в пошуковий кортеж. Співставлення з кортежами в репозиторії проводиться поаспектно (тобто співставляється частина кортежу, яка відповідає поточному аспекту, з відповідною частиною кортежу в ПОБ, який у даний момент співставляється)

Позначимо операцію трансформування графа запиту  $G_{ПОЗ} \subseteq G_A = (V_A, D_A)$  (граф аспекту онтології домену) у пошуковий кортеж  $X_{ПОЗ}$  як  $F_{ПОЗ}: G_{ПОЗ} \rightarrow X_{ПОЗ}$ , де  $G_{ПОЗ} = (V_{ПОЗ} \subseteq V_A, D_{ПОЗ} \subseteq D_A, \alpha)$  - помічений граф (граф запиту),  $V_{ПОЗ}, D_{ПОЗ}$  - множина активованих вершин та дуг в запиті, а  $X_{ПОЗ}$  - кортежу запиту на мові МОК (або пошуковий кортеж), який представляє опис компоненти.  $F_{ПОЗ}$  – функція, яка побудована на базі функції  $\alpha$  з обмеженою областю дії.  $\alpha$  – це функція відповідності вершини графа концепту онтології.

Оскільки підхід до побудови ПоЗ та ПоБ аналогічні наведемо тільки правила побудови багатоаспектного ПоБ на мові МОК.

Побудова кортежу багатоаспектного образу компоненти на мові МОК:

1. Опис компоненти на МОК представляє комбінацію ПоБ компоненти за декількома аспектами.

2. Текст опису на МОК починається з назви компоненти та домену, в якому ця компонента атестується.

3. Опис компоненти на МОК має декілька розділів, кожний з яких відповідає аспекту. Опис за кожним аспектом починається з назви аспекту та після двокрапки векторний опис відповідного онтологічного фрагменту.

4. Назвам тегів на МОК відповідає назва концептів онтології верхнього рівня (в графі родові вершини породжують гілки аспекту, тобто це вершини другого рівня ієрархії графа).

5. Вкладені теги можуть фігурувати в опису та представляти також верхній рівень родових концептів.

6. В опис ПоБ, відповідний компоненті в поточному аспекту, вноситься вектор концептів, які відповідають активованим вершинам графа ПоБ. Послідовність включення концептів до тегу опису аспекту зумовлено алгоритмів обходу графа в ширину.

7. Означення. Відношення, яке відповідає дузі графа, яка поєднує дві активовані вершини графа назвемо активованим відношенням ПоБ. Також до вектору концептів можуть бути вбудовані відповідні назви відношень між концептами (також у послідовності обходу графа в ширину), якщо відношення поєднує активовані для ПоБ відношення.

8. Сформований таким чином опис компоненти на мові МОК зберігається у відповідному розділі репозиторія системи.

9. Зазначимо, що при виконанні обов'язкових процедур, які на протязі еволюціонування домену будуть підтримувати коректність та цілісність напрацьованих знань в репозиторії системи, в будь-який момент часу образ компоненти, який зберігається в репозиторії, може бути візуалізовано. Під процесом візуалізації ПоБ компоненти будемо розуміти процес активації у модифікованому (за вимогами еволюціонування домену в часі) графі ПОБ відповідних вершин.

10. Засіб візуалізації ПоБ значно розширює можливості візуального співставлення графічних образів компонент та

підтримує прийняття рішення про доцільність використання тієї чи іншої компоненти в цільовій розробці.

### 2. Розширений апарат критеріїв релевантності в онтологічній системі

Задача співставлення [8] може ставитися як співставлення пошукового запиту і пошукового образу ПК з метою вибору останніх або як співставлення пошукових образів двох ПК з метою порівняння їх властивостей. Одна із таких методик запропонована у [9 - 11] для обмежених типів ПК (функції, які співставляються за типами вхідних та вихідних параметрів).

Нагадаємо, що для співставлення використовуються наведені далі структури. До них відносимо:

1. пошуковий образ ПК (ПоБ), який представлено як конкатенацію фрагментів, кожен з яких визначає певний аспект та перелік відповідних йому дескрипторів. Впорядкованість переліку концептів у фрагменті аспекту та послідовність фрагментів аспектів не визначається. Для атестації різних ПК використовується довільний склад аспектів;

2. пошуковий запит (ПоЗ), структура якого ідентична структурі пошукового образу, склад відображених у ньому аспектів також довільний;

3. набір онтологій  $O(A_i)$ , відповідних тим аспектам  $A_i$ , які підтримує ПК.

Процес співставлення відбувається як поаспектне порівняння відповідних розділів ПоЗ та ПоБ, а саме:

1. Перевіряється наявність аспектів, зазначених у ПоЗ, у складі аспектів ПоБ (відповідних аспектів).

2. Для знайдених відповідних аспектів перевіряється наявність концептів ПоЗ (та значущих відношень) у складі ПоБ.

Позначимо пошуковий образ компоненти як

$$D_{pk} = \cap D(A_i), \text{ де } 1 \leq i \leq N, A_i \subseteq O(D).$$

Тут  $D_{pk}$  – пошуковий образ однієї ПК,

$A_i$  – аспект, по якому атестується ПК,

$D(A_i)$  – множина концептів, відповідних  $A_i$ ,

$O(D)$  – множина всіх онтологій аспектів в домені.

Відповідно пошуковий запит подамо так:

$Q = \bigcap Q(A_j)$ , де  $1 \leq j \leq k$ ,  $A_j \subseteq O(A_i)$ , а  $Q(A_j)$  – множина концептів відповідних  $A_j$ .

Тоді будемо позначати співставленням  $Q R C_{ПК}$  пошукового образу ресурсу  $C_{ПК}$  з пошуковим запитом  $Q$  :

де  $R$  – предикат співставлення (критерій релевантності).

$R: Q, C_{ПК} \rightarrow bool$ ,

$R = \bigcap R_j$ , де  $1 \leq j \leq k$ , де  $k$  – кількість аспектів, за якими був побудований пошуковий запит  $Q$ , а  $R_j$  – предикат співставлення за аспектом  $A_j$ .

Тобто при створенні пошукового образу шукач повинен починати атестацію з того аспекту, який він вважає найзначнішим для пошуку. І потім поступово переходити к менш значним за його розумінням.

Пошук точно релевантних запити ресурсів може не дати результатів. Тоді застосовуються стратегії послаблення критерію релевантності.

Підхід багатоаспектної атестації ПК при побудові ПоЗ та ПоБ дозволяє розширити коло співставлень з метою виявлення адекватних ПК в репозиторії. Для проведення найбільш результативного пошуку адекватної вимогам ПК вводяться різні за типом предикати ослаблення точного співставлення. Кожний з таких предикатів прислуговує для проведення послабленого співставлення між побудованим користувачем пошуковими образами та пошуковим запитом. У роботі оговорюються підходи до побудови пошукового запиту в системі з використанням комбінації різних за типом предикатів з метою підвищення ефективності пошуку в репозиторії.

Типові прийоми послаблення або посилення критеріїв пошуку в сучасних ІПС – це скорочення або розширення пошукових умов, що, як правило, виконується вручну користувачем. Онтологічний принцип керування роботою репозиторію надає можливості обирати стратегії зміни критеріїв релевантності, керуючись базою знань репозиторію, зокрема тими відношеннями,

які підтримуються онтологіями аспектів. При цьому для ряду стратегій змін пошукові вирази можуть корегуватися автоматично. Серед таких визначено стратегії послаблення критеріїв релевантності, до яких може бути віднесено зміна аспектів, за якими ведеться пошук, зміна набору концептів, зміна синонімів на канонічні терміни в словнику онтології, або навпаки, зміна стратегії побудови ПоЗ (вказування тільки родові вершини або значення для встановлення релевантності). До критеріїв, які дозволяють посилити співставлення віднесемо наступні введення до предикату співставлення за специфічним відношенням, введення конкретизації (тобто пошук тільки серед нащадків вказаних у запиті родових концептів), додавання до атестації нових аспектів, додавання до атестації нових концептів.

*Означення.* Будемо називати співставлення за одним аспектом простим співставленням.

*Означення.* Відповідно, співставлення за декількома аспектами назвемо складним. Зрозуміло, що декілька простих співставлень складають одне складне співставлення.

Наведемо в таблиці перелік основних послаблених та підсилюючих предикатів співставлення. Для кожного визначимо предикатні символи і зміст.

Зазначимо, що на практиці послаблення може здійснюватися вибірково за аспектами. Тобто складний предикат може включати декілька різних простих послаблених предикатів.

Крім вище зазначених типів предикатів послаблення точного співставлення можуть бути введені предикати, що є специфічними для окремих аспектів і не можуть використовуватися для інших.

Різні варіанти трансформацій наведених предикатів відповідають ряду обмежень пошукового запиту, реалізація яких надасть більш релевантні ПК. Будемо називати таке ітераційне співставлення трансформаційним.

Такий трансформаційний апарат може використовуватися з метою виявлення груп компонентів, придатних для повтор-

Таблиця. Предикатні символи та зміст предикатів співставлення образів ресурсів в багатоаспектній системі

№ п/п	Назва	Зміст	Предикатний символ
1	Ігнорування найменш значного за думкою експерта аспекту в запиті	Складний ПоЗ, який складається з декількох простих (атестації по одному аспекту), послабляється шляхом відкидання незначного за думкою експерта аспекту з ПоЗ	$R_{\text{мінус асп}}$
2	Відкидання найбільш не релевантного за підказкою системи	Таке послаблення має сенс після підказки системи, який з представлених в ПоЗ аспектів є найбільш не релевантним критерію	$R_{\text{релев асп}}$
3	Ігнорування в одному аспекті певної кількості дескрипторів, які відмічені в ПоЗ	Це ослаблення проводиться в одному аспекті в простому запиті. При цьому видаляються менш значні за думкою експерта індексатора ПК концепти	$R_{\text{дескр}}$
4	Ігнорування декількох дескрипторів в складному запиті	Це відкидання дескрипторів декількох простих запитах одночасно. Тобто корегування всього складного запиту	$R_{2 \text{ дескр}}$
5	Узагальнене співставлення	Під узагальненим співставленням будемо розуміти співставлення, коли замість означеного в ПоЗ дескриптора для пошуку використовуються його батьки в ієрархії онтології аспекту	$R_{\text{узагальнене}}$
6	Звужене співставлення	Відповідно, при звуженому співставленні допускається, коли замість визначеного в ПоЗ дескриптора можна використати дескриптори, що конкретизують вказаний у ієрархії онтологій аспект	$R_{\text{звужене}}$
7	Співставлення з радіусом пошуку	При цьому типі співставлення задається радіус пошуку, який можна ігнорувати при співставленні вершин ПоЗ з ПоБ компоненти	$R_{\text{радіус}}$
8	Співставлення по відношенню	При цьому типі співставлення вважаються релевантними ті вершини, які безпосередньо зв'язані вказаним відношенням з поміченими в ПоЗ вершинами	$R_{\text{відн}}$
9	Співставлення по групі відношень	При цьому типі співставлення вважаються релевантними ті вершини, які безпосередньо зв'язані відношеннями із групи вказаних при виборі предикату з поміченими в ПоЗ вершинами	$R_{2 \text{ відн}}$

ного їх використання, аналізу, проглядання, а також з метою фільтрації бази знань домену та репозиторія образів компонент. Також бачиться перспективним використання різних типів предикатів співставлення для структуризації бібліотеки з метою створення навігаторів прискореного пошуку.

Зазначимо, що апарат предикатів співставлення потребує використання метрик виконаного співставлення.

Запропонуємо метрику дії критеріїв співставлення, яка базується на встановленні зв'язаних шляхів між концептами онтології, вага яких залежить від типу від-

ношень між концептами на шляху. Специфіка цієї метрики в тому, що вона дозволяє встановити не тільки оцінку співставлення за відповідністю заданому критерію, а за бажанням користувача оцінку "сили" зв'язку між концептами ПоБ та концептами ПоЗ. Ця "сила" визначається числом, яке є "вагою" зв'язку між концептами онтології. Для кожного типу базового відношення в онтології воно постійне. За бажанням експерта для специфічних відношень може бути також вибрано числове значення (або не встановлювати його зовсім для відношень, які за думкою експерта є не типовими та рідко використовуваними в аспекті).



*Означення.* Відношення, яке має відповідну числову мітку, будемо називати зв'язком.

Два однакових за типом послідовних зв'язків дають новий зв'язок. Таким чином встановлюється нова вага шляху, яка для шляху визначається по-різному для різних типів відношень. Визначимо правила встановлення ваги для шляху, який породжується двома послідовними зв'язками в залежності від типу зв'язку. Встановимо ці правила для набору базових відношень:

1) для зв'язку “асоціація” сила (вага) нового зв'язку, який породжується двома послідовними зв'язками, дорівнює максимальній силі з цих двох;

2) для зв'язку, який породжений послідовністю двох зв'язків “спеціалізація”, сила результуючого зв'язку дорівнює максимальній силі з цих двох;

3) для зв'язку “агрегація”, сила результуючого зв'язку дорівнює мінімальній силі з цих двох;

4) для зв'язку “вид” сила (вага) нового зв'язку, який породжується двома послідовними зв'язками, дорівнює мінімальній з сил обох зв'язків.

Якщо слід встановити силу шляху, який складається з різних типів відношень, то вона дорівнюватиме сумі ваг різних відношень.

Якщо зв'язаний зв'язок переривається, то результуючої ваги шляху не існує.

Якщо всі зв'язки в шляху є зв'язками одного типу, причому це відношення охарактеризоване як транзитивне, будемо вважати, що кінцеві концепти шляху знаходяться у цьому відношенні (наприклад, віс відношення шляху “спеціалізація”, тоді результуюче відношення шляху буде “спеціалізація”).

*Означення.* Позитивним зв'язком між концептами назвемо зв'язок, при якому результуюче відношення між концептами в онтологічній ієрархії існує та не перевищує поріг, встановлений експертом домену.

Якщо в шляху існує хоча б один зв'язок, який не має “ваги”, то результую-

чого відношення шляху не існує та концепти не поєднані позитивним зв'язком.

Для того, щоб підрахувати силу шляху, який включає асоціації різних типів, необхідно в першу чергу знайти сили його підшляхів. Задача алгоритму – знайти всі шляхи та всі ваги шляхів між концептом ПоЗ та концептами, які складають ПоБ компоненти. Якщо знайдені “ваги” шляхів не перевищують порогового значення, то такі концепти вважаються адекватними запиту. Якщо не знайдено жодного концепту, який є адекватний запиту, то таке ПоБ не входить до результату запиту.

Наведемо операції онтологічної системи по проведенню співставлення при впровадженні пошуку та співставлення ПоБ з ПоЗ.

У випадку, коли користувач бажає задіяти наведений апарат зміни критерію співставлення за допомогою вагової оцінки, він повинен:

- 1) встановити поточний аспект;
- 2) включити режим побудови ПоЗ;
- 3) в діалоговому режимі встановити (або залишити ті, які встановлені по замовченню) числа для ваг відношень, які він вважає за потрібне використати при поточному співставленні;

4) у випадку, якщо відношення ігнорується при використанні вагової метрики він не має ваги та таким чином перериває зв'язаний шлях;

5) у випадку, коли користувачем встановлюється вага рівна 0, вона не впливає на оцінку шляху, але і не руйнує його позитивності;

6) користувач має встановити поріг, який дозволить у подальшому розширити побудоване первинне ПоЗ до визначеної межі. Такий параметр вибирається експериментально та може бути змінений в подальшому у випадку отримання результату, який не задовольняє запитувача ПК;

- 7) побудувати первинний ПоЗ;
- 8) обрати один з типів предикатів співставлення (наприклад, співставлення за одним типом відношення);

9) встановити ваги предикатів співставлення (наприклад, співставлення за одним типом відношення);

- 7) побудувати первинний ПоЗ;
- 8) обрати один з типів предикатів співставлення (наприклад, співставлення за одним типом відношення);

9) встановити ваги предикатів співставлення (наприклад, співставлення за одним типом відношення);

9) виконати активізацію пошуку;

10) проглянути отримані результати, тобто назви та ПоБ релевантних ПоЗ компонент;

11) у випадку нерелевантності результатів вимогам користувача можливе ітераційне повторення дій (починаючи з пункту 3 або 6) з подальшим корегуванням порогу розширення (або звуження) адекватної області по відношенню до ПоЗ до моменту отримання задовольняючого результату.

На базі знайдених у процесі роботи подібного алгоритму зв'язків для різних відношень між концептами в онтологічній системі може бути побудована узагальнена числова метрика “звішеної” відстані між ПоБ та ПоЗ програмної компоненти.

Така метрика дозволить встановлювати ефективність того чи іншого предикату співставлення при порівнянні ПоБ та ПоЗ компонент на базі відношення слабкого чи сильного зв'язку між концептами ПоБ та ПоЗ компонент.

Назвемо відношення між двома концептами  $C_i$  та  $C_j$   $R_{звішене}$ .

Будемо вважати, що концепт  $C_i$ , який приймає участь в ПоБ компоненти релевантний за вагою шляху (тобто знаходиться у зв'язаному відношенні) концепту  $C_j$ , який приймає участь в ПоЗ, якщо  $C_i$  має результуючий зв'язок з  $C_j$  ( $positive(C_i, C_j)$ ).

$R_{звішене}(C_i, C_j) \Leftrightarrow (\exists Iij: Iij = positive(C_i, C_j))$ . В цьому означенні  $Iij$  – числове значення, яке не більше встановленого порогового значення для даного типу відношення.

Будемо вважати, що концепт  $C_j$ , який приймає участь в ПоБ компоненти релевантний за вагою шляху (тобто знаходиться у зв'язаному відношенні) ПоЗ в аспекті онтології  $A$ , якщо в ПоЗ існує  $C_i$ , яке має позитивний зв'язок з  $C_j$  ( $Iij = positive(C_i, C_j)$ ).

Будемо вважати, що ПоБ компоненти релевантний за вагою шляху (тобто знаходиться у зв'язаному відношенні) до ПоЗ в аспекті онтології  $A$ , якщо для будь-якого концепту з  $C_i$  з ПоБ існує  $C_j$  з ПоЗ,

такий що є позитивний зв'язок  $C_i$  з  $C_j$  ( $Iij = positive(C_i, C_j)$ ).

Методики комбінування різних за типом предикатів співставлення та методики встановлення зв'язаного концептуального зв'язку дозволяють виявляти ті компоненти, які відповідають запиту за встановленим предикатом (тобто критерієм релевантності) та одночасно ті ПоБ компонент в репозиторії, які пов'язані з поточним ПоЗ “сильним” позитивним зв'язком. Нехай запит ПоЗ складається з кортежу концептів онтології  $(C_1, C_2, \dots, C_i, \dots, C_n)$ . Нехай  $R$  – один з предикатів, які наведені в таблиці вище.

Тоді запит  $Q$  (ПоЗ,  $P$ ,  $R$ ,  $A$ ) = {ПоБ<sub>*i*</sub> | ПоБ<sub>*i*</sub>  $\subseteq A$ ,  $positive(ПоЗ, ПоБ<sub>*i*</sub>) \leq P$ ,  $позR_{Поб} = true$ } , де  $Positive(ПоЗ, P)$  – сила позитивного зв'язку за встановленим пороговим значенням  $P$  релевантності шляху між значенням у запиті ПоЗ та поточно розглядуваним ПоБ<sub>*i*</sub>, який міститься в репозиторії,  $A$  – аспект онтологічної ієрархії домену.

Саме потужність множини  $Q(ПоЗ, P, R, A)$  є визначальною для шукача ПК для прийняття рішення про продовження сесії пошукових запитів, або її зупиненні. Також важливим є те, що результат запиту формується як комбінація результатів за кожним з аспектів. Значимість результату за кожним з аспектів та те, наскільки вагомим є результат за окремими аспектом дозволяє прийняти рішення про доцільність використання ПК у новій розробці. Нагадаємо, що ті з предикатів, які не є важливими на даний момент пошукової сесії, шукач має змогу відкинути вже на етапі підбору відповідного предикату співставлення.

## Висновки

На базі побудованої у рамках обраного домену онтології пропонується метод аналітичного поаспектного співставлення різних за типом програмних компонент. Він полягає у використанні предикатів співставлення з метою проведення поаспектного аналізу інформаційних моделей ПК.

Базис цього підходу становлять предикати співставлення, які пропонується використовувати при проведенні транзакцій співставлень з метою виявлення адекватної вимогам користувача ПК.

Введені предикати співставлення дозволяють оцінити подібність між двома образами ПК. Для проведення найбільш результативного пошуку адекватної вимогам ПК вводяться різні за типом предикати співставлення. Кожний з предикатів прислуговує проведенню точного або послабленого співставлення між побудованим користувачем пошуковими образами та пошуковим запитом, які будуються відповідно індексатором та шукачем ПК в онтологічній системі.

Можливість варіювання типом використовуваних у запиті предикатів співставлення є перевагою даного методу. Рішення про використання в запиті того чи іншого предикату приймається шукачем бази після проведення аналізу ефективності результатів попередніх пошукових запитів. Шукач має можливість уточнити чи послабити запити в залежності від кількості та якості тих ПК, які пропонуються йому для повторного використання в результаті проведеного попереднього пошукового запиту. Таким чином поетапно будується гнучкий пошуковий запит для пошуку адекватних вимогам шукача компонент.

Комбінації різних за типом предикатів та одночасно використання метрики “сильного” позитивного зв’язку між ПоЗ та ПоБ дозволить отримати більш точний результат при впровадженні пошукової сесії у багатаспектній онтологічній системі.

У подальшому планується розширити коло використовуваних при побудові запиту предикатів співставлень, а також провести аналіз ефективності використання конкретних предикатів у запитах. Також планується робота по проведенню співставлення ефективності різних комбінацій предикатів при побудові запитів. Цікаво виявити залежність ефективності того чи іншого запиту від структури онтології, на базі якої він будується.

1. *Бабенко Л.П., Полянничко С.Л.* Онтологічні моделі опису готових ресурсів у розробці програм. Пр. міжнар. конф. УКРПРОГ. - 2004. - № 2-3. - С.173-179.
2. *Gomes-Perez A., Fernandez-Lopez M., Corcho O.* *Ontological Engineering // Springer –Verlag London Limited.* - 2004. – 403 с.
3. *Бабенко Л.П.* Проблемы повторного использования в программной инженерии. Кибернетика и системный анализ.-1999. - №2. - С. 155-166.
4. *Robert B. France, Sudipto Ghosh.* *A UML-Based Pattern Specification Technique// IEEE Transactions on Software Engineering.* - 2004. - v.3. - №3. - P. 193-206.
5. *Грищенко В.М.* Методи еволюції програмних компонентів для їх повторного застосування. Труды міжн. конф. По програмуванню УКРПРОГ. - 2004. - № 2-3. - С. 215-222.
6. *Бабенко Л.П., Полянничко С.Л.* Подходы к аттестации адаптивных возможностей программных компонент// Там же. – 2001. – N 1-2. – С. 35-41.
7. *Как программировать на XML.* – М.: “Бином”, 2001. – 933 с.
8. *Полянничко С.Л.* Предикати співставлення для порівняння та аналізу компонент// Проблеми програмування. – 2002. – N 1-2. – С. 117-120.
9. *Zaremsky A.M., Wing J.M.* *Specification Matching of Software Components//ACM Software Eng. Notes.* - 1995. - August. - P. 6-16.
10. *Zaremsky A.M., Wing J.M.* *Signature Matching; A Tool for Using Software Libraries// ACM Transaction on Software Engineering and Methodology.* - 1995. - 4, N2. - P. 146-170.
11. *Zaremsky A.M., Wing J.M.* *Specification Matching; A Tool for Using Software Libraries// ACM Transaction on Software Engineering and Methodology.* - 1997. - 6, N4. - P. 333-369.

Отримано 27.11.2006

***Про автора:***

*Полянничко Софія Леонідівна*

м.н.с. відділу №14

“Програмна інженерія”

***Місце роботи автора:***

Інститут програмних систем

НАН України.

04053, Київ, вул. Артема, д.59/65, кв. 63,

тел. 486-71-50,

e-mail: sonypol@rambler.ru,

sonypol@voliacable.com