

DOI <https://doi.org/10.15407/usim.2019.03.023>  
УДК 004.318

**О.В. ПАЛАГІН**, академік НАНУ, д-р техн. наук, професор, заст. директора,  
Інститут кібернетики ім. В.М. Глушкова НАН України,  
просп. Глушкова, 40, Київ 03187, Україна,  
[palagin\\_a@ukr.net](mailto:palagin_a@ukr.net)

**М.Г. ПЕТРЕНКО**, д-р техн. наук, провідний науковий співробітник,  
Інститут кібернетики ім. В.М. Глушкова НАН України,  
просп. Глушкова, 40, Київ 03187, Україна,  
[petrng@ukr.net](mailto:petrng@ukr.net)

## ПРО ДЕЯКІ ОСОБЛИВОСТІ ПОБУДОВИ ОНТОЛОГІЧНИХ МОДЕЛЕЙ ПРЕДМЕТНИХ ОБЛАСТЕЙ

*Розглянуто особливості різних моделей побудови онтологій предметних областей. Виконано їх порівняльний аналіз і показано, що перспективним напрямком є побудова веб-онтологій в середовищі Protege. Наведено приклад розробки такої онтології в інтегрованому середовищі онторедактора Protege, механізму міркувань Pellet і SPARQL-процесора з відповідними до неї запитами.*

**Ключові слова:** *описувальна логіка, онтологія, модель онтології, концептуалізація, онтограф, Protege.*

### Вступ

Для терміна “онтологія” поширеними є два сенси:

- в філософському сенсі онтологія вивчає природу і структуру речей як таких, незалежно від їх фактичного існування;
- другий сенс найбільш поширений в інформатиці як особливий вид інформаційного об'єкта.

Важливість дійсного існування сутностей (які розглядаються в цій онтології) прагматична [1–3]. Онтології в онтологічному інжинірингу є засобом формального моделювання структури системи, тобто відповідних об'єктів або суб'єктів і відношень між ними, які належать відповідній предметній області (ПдО) і використовуються для вирішення практичних завдань.

Далі термін “онтологія” буде розглядатися в другому сенсі, з точки зору онтологічного інжинірингу.

Прикладом такої системи вибрано деяку виробничу компанію з її виробничими об'єктами, співробітниками і виробничими відносинами між ними. Інженер зі знань аналізує відповідні сутності та класифікує їх на поняття і відношення, представлені відповідно унарними і бінарними (в загальному випадку  $n$ -арними) предикатами. Базова структура онтології заснована на процедурах узагальнення/спеціалізації ієрархії понять, тобто таксономії. Припустимо, що авторів цікавить сукупність задач, пов'язаних з людськими ресурсами компанії. Тоді сутності *Співробітник*, *Майстер* і *Робочий* є релевантними для вирішення сукупності практичних задач, пов'язаних з людськими ресурсами компанії. Перше поняття є суперпоняттям (узагальнюючим, родовим поняттям) для інших двох понять. Відношення “працює-з” є релевантним відношенням між співробітниками (не тільки робочими). Тоді конкретна людина,

яка працює в компанії, буде екземпляром (індивідом) відповідного поняття.

Метою даної статті є розгляд найбільш відомих інструментів формального опису онтологій ПдО, їх порівняльний аналіз і вибір формальної мови і онторедатора для практичного використання.

## Визначення онтології та її складових

Початкове визначення терміна “онтологія” [1] зазнало ряд змін, і в підсумку в співтоваристві онтологів дійшли до наступного визначення: “Онтологія — це формальна, явна специфікація загальної (яка колективно використовується) концептуалізації” [2–5]. Далі коротко розглянемо три складові наведеного визначення (концептуалізація, формальна, явна специфікація і чому загальна (яка колективно використовується) концептуалізація так важлива) з урахуванням того, що вони докладно обговорювалися в [2–7].

**Концептуалізація.** В [2, 3] наведено таке визначення поняття “концептуалізація”, в якому стверджується: «Сукупність формально представлених знань ПдО заснована на концептуалізації: об’єктах, поняттях та інших сутностях, які, ймовірно, існують в деякій ПдО, що розглядається, і відношень між ними. Концептуалізація — це абстрактне, спрощене уявлення про світ, яке розглядається з деякою метою. Кожна база знань, знання-орієнтована система або знання-орієнтований агент дотримуються визначеної концептуалізації, явно або неявно». Незважаючи на складну ментальну природу поняття “концептуалізація”, в [7] його визначення засноване на простому математичному поданні: екстенціональній реляційній структурі.

**Екстенціональна реляційна структура** (концептуалізація згідно [7]) є кортежем  $(D, R)$ , де:

- $D$  — множина понять  $X$  предметної області, названа областю визначення;
- $R$  — множина відношень на  $D$ .

Зауважимо, що в цьому визначенні елементами множини  $R$  є звичайні математичні

відношення на  $D$ , тобто множини упорядкованих кортежів елементів з  $D$ . Таким чином, кожен елемент  $R$  є екстенціональним відношенням, що відображає конкретний стан світу з елементами з  $D$ .

Незважаючи на простоту визначення екстенціонального поняття “концептуалізація”, в дійсності вона слабо підходить для вирішення практичних задач, головним чином тому, що значною мірою залежна від конкретного стану ПдО.

Суть в тому, що екстенціональні відношення, які належать  $R$ , відображають конкретний стан фрагмента ПдО. Отже, необхідно зосередитися на смисловому значенні основоположних понять, що не залежать від конкретного стану ПдО (або згідно Кріпке, можливого світу): наприклад, значення відношення *працює-з* полягає в тому, як взаємодіють два конкретних робочих. На практиці розуміння такого значення має на увазі наявність правил для прийняття рішень, дотримання різних моделей поведінки, незалежно від того, чи працюють ці дві людини чи ні.

Припустимо, в прикладі для двох робочих відношення «*працювати-з*» означає, що:

- обидва заявляють, що мають одну і ту ж мету;
- обидва роблять що-небудь для досягнення цієї мети.

Тоді значення «*працювати-з*» можна визначити як *функцію*, яка для кожного глобального поведінкового контексту, що включає всю область визначення, дає список пар робочих, які фактично працюють разом в цьому контексті. Зворотна дія цієї функції *обгрунтовує* значення поняття в конкретному стані ПдО. Узагальнюючи цей підхід і абстрагуючись від часу заради простоти, припустимо, що інтенціональне відношення (або концептуальне відношення згідно [8]) є функцією з множини максимальних станів ПдО (глобальні поведінкові контексти в цьому випадку) в екстенціональні відношення. Це загальний спосіб вираження змісту понять, який бере свій початок від Карнапа [9], прийнятий і розширений в семантиці Монтегіу [10].

В [6] представлено наступне визначення *інтенціонального або концептуального відношення*. Нехай  $S$  — довільна система,  $D$  — довільна множина елементів, які розрізняються в  $S$ ,  $W$  — множина станів ПдО для  $S$ . Кортеж  $\langle D, W \rangle$  називається областю визначення для  $S$ , оскільки він інтуїтивно фіксує простір змін області міркування в  $D$  відносно можливих станів  $S$ . *Інтенціональне відношення* (або концептуальне відношення)  $\rho^n$ -арності  $n$  на  $\langle D, W \rangle$  — це повна функція  $\rho^n : W \rightarrow 2^{D^n}$  з множини  $W$  в множину всіх  $n$ -арних (екстенсіональних) відношень на  $D$ .

Визначивши концептуальне відношення, далі представлено визначення загальної концептуалізації (структури інтенсіональних відношень).

Структура інтенсіональних відношень (або загальна концептуалізація, далі просто концептуалізація) являє собою трійку  $C = (D, W, \mathfrak{R})$ , де:

- $D$  — область дискурса;
- $W$  — множина можливих світів;
- $\mathfrak{R}$  — множина концептуальних відношень

на доменному просторі  $\langle D, W \rangle$ .

**Формальні, явні специфікації.** У практичних додатках необхідно використовувати мову для позначення елементів концептуалізації. У цьому випадку говорять, що мова  $L$  фіксує концептуалізацію. Далі необхідно в явному вигляді визначити концептуалізацію і тут проявляється роль онтологій як «явних специфікацій концептуалізації». В принципі, таку концептуалізацію можна чітко визначити двома способами: екстенсіонально й інтенсіонально. У цьому прикладі екстенсіональне визначення концептуалізації потребує перерахування розширень кожного (концептуального) відношення для всіх можливих світів. У загальному випадку, більш ефективний спосіб визначення концептуалізації полягає в тому, щоб зафіксувати мову і *обмежити інтерпретації такої мови інтенсіонально*, за допомогою відповідних аксіом (в [9] названих *смысловими постулатами*). Наприклад, можна взяти прості аксіоми, які стверджують, що відношення *підпорядковується* — асиметричне і нетранзитивне, а відношення *працює-з* симетричне, нереклексивне і нетранзитив-

не. І *онтологія* — це набір таких аксіом, тобто логічна теорія, розроблена для вибору передбачуваних моделей, які відповідають певній концептуалізації, і виключення непередбачуваних моделей. Результатом буде *приблизна* специфікація концептуалізації: будуть відібрані кращі передбачувані моделі і виключені непередбачувані моделі.

Аксіоми для інтенсіонального й екстенсіонального специфікування концептуалізації можуть бути представлені на *неформальній* або *формальній* мові  $L$ . Як вимагається у визначенні онтології (і згідно [2, 4, 5]), явна специфікація повинна бути формальною. Термін «формальний» передбачає, що вирази на цій мові повинні бути такі, які читаються машиною, тому природна мова виключається. Далі слід представити всі викладені поняття більш формальним способом.

Нехай мова  $L$  є (варіантом) логічної мови першого порядку, причому словник  $V$  складається з множини констант і предикатних символів (функціональні символи не розглядаються). Введемо поняття *онтологічного зв'язування* (*ontological commitment*), розширивши стандартне поняття екстенсіональної структури першого порядку до інтенсіональної структури першого порядку.

**Онтологічне зв'язування** для  $L$  подається кортежем  $K = (C, \hat{I})$ , де (інтенсіональна функція інтерпретації) — повна функція  $\hat{I} : V \rightarrow D \cup \mathfrak{R}$ , яка відображає кожен словниковий символ з  $V$  чи в елемент  $D$ , чи в інтенсіональне відношення, яке належить множині  $\mathfrak{R}$ .

Тепер видно, що визначення *онтологічного зв'язування* розширює звичайне (екстенсіональне) визначення «сенсу» для словникових символів в інтенсіональному випадку, замінюючи поняття моделі поняттям концептуалізації.

Показано, що поняття *онтологічне зв'язування* є розширенням стандартного поняття моделі. Останнє поняття — екстенсіональна оцінка сенсу, перше — інтенсіональна оцінка сенсу. Слід з'ясувати, який взаємозв'язок між ними? Звичайно, як тільки буде визначено інтенсіональний сенс словника через його онтологічне зв'язування, також будуть пев-

ним чином обмежені і його моделі. Введемо поняття *передбачуваної моделі* щодо певного онтологічного зв'язування.

Нехай  $C = (D, W, \mathfrak{R})$  — концептуалізація,  $L$  — логікова мова першого порядку з словником  $V$  і онтологічним зв'язуванням  $K = (C, \hat{I})$ . Модель  $M = (S, I)$ , де  $S = (D, R)$ , називається передбачуваною моделлю  $L$  на  $K$ , якщо:

- для всіх константних символів  $c \in V$  маємо  $I(c) = \hat{I}(c)$

- існує такий світ  $w \in W$ , що для кожного предикатного символу  $v \in V$  дійсно є таке інтенціональне відношення  $\rho \in \mathfrak{R}$ , що  $\hat{I}(v) = \rho$  та  $I(v) = \rho(w)$ . Множина  $I_K(L)$  всіх моделей  $L$ , спільних з  $K$ , називається множиною передбачуваних моделей  $L$  на  $K$ .

Наведена умова 1 вимагає, щоб відображення константних символів в елементи області дискурсу було ідентичним. Умова 2 стверджує, що повинен існувати такий світ, що кожен предикатний символ відображається в інтенціональне відношення, значення якого для цього світу збігається з екстенціональною інтерпретацією такого символу. Це означає, що передбачувана модель буде описом цього світу. Визначивши поняття передбачуваних моделей, можна уточнити роль онтології, яка розглядається як логікова теорія і призначена для обліку передбачуваного значення словника, який використовує логікову мову.

**Онтологія.** Нехай  $C$  — концептуалізація,  $L$  — логікова мова зі словником  $V$  і онтологічним зв'язуванням  $K$ . Онтологія  $O_K$  для  $C$  зі словником  $V$  і онтологічним зв'язуванням  $K$  є логіковою теорією, яка складається з набору формул із  $L$ , побудованих таким чином, що множина його моделей наближається як можливо точніше до множини передбачуваних моделей  $L$  по  $K$ .

Отже, можна зробити висновок, що «ідеальна» онтологія — це така онтологія, чії моделі точно збігаються (по модулю ізоморфізмів) з передбачуваними моделями. Однак не все так просто: навіть «прекрасна» онтологія може не точно визначити свою цільову концептуалізацію, якщо її *словник* і *область дискурсу* не підібрані відповідним чином. Причина

полягає у відмінності між *логіковим* поняттям *моделі* і *онтологічним* поняттям *можливого світу*. Перше поняття є комбінацією присвоювань з абстрактних реляційних структур (побудованих на області дискурсу) з елементами словника; друге поняття є комбінацією дійсних (які спостерігаються) станів ПдО конкретної системи. Звичайно, кількість можливих моделей залежить як від розміру словника, так і від розширення області дискурсу, які вибираються більш-менш довільно, виходячи з того, що, очевидно, має відношення до цільового призначення. Навпаки, число станів ПдО залежить від змінних, які спостерігаються, навіть тих, які на перший погляд вважаються такими, що не мають відношення до цільового призначення. Слід підкреслити, що ступінь, з якою онтологія визначає концептуалізацію, залежить від:

- обсягу області дискурсу;
- представницької множини термінів, які увійшли у словник;
- аксіоматизації.

В свою чергу, аксіоматизація залежить від мовної виразності.

**Спільне (колективне) використання онтологій.** Перші визначення «онтологій» не розглядали аспект спільного використання онтологій [3, 11] і тільки пізніше його було введено в [4]. Дійсно, можна з упевненістю стверджувати, що неможливо зробити концептуалізації, що спільно використовуються, які є окремими випадками для кожного індивіда.

Що можна зробити таким, які спільно використовуються, для наближень концептуалізацій, заснованих на обмеженій множині прикладів, і показують фактичні стани, при яких мають місце певні концептуальні відношення (наприклад, фактичні ситуації, що показують випадки, коли виникає відношення *працює-з*). Крім простих прикладів також можна зробити смислові постулати, які спільно використовуються, тобто явні формальні обмеження. Однак такі визначення припускають взаємну згоду на примітивні терміни, які використовуються в цих визначеннях. Оскільки смислові постулати не можуть в повній мірі



характеризувати онтологічне зв'язування за допомогою примітивних термінів, слід врахувати, що узагальнення концептуалізацій в кращому випадку є частковим. Для практичного використання онтологій зрозуміло, що без мінімального загального онтологічного зв'язування (спільного використання, наприклад, веб-онтологій) від зацікавлених в онтології сторін, переваги онтології обмежені.

Причина в тому, що онтологія формально визначає доменну структуру при обмеженні, що розробник правильно розуміє примітивні терміни. Інакше, онтологія може виявитися марною, якщо вона використовується таким чином, що суперечить загальному онтологічному зв'язуванню. Будь-яка онтологія завжди буде менш повною і менш формальною, ніж можливо теоретично. Ось чому важливо, щоб онтології, призначені для підтримки великих проектів, були обґрунтованими в тому сенсі, що основні примітиви, на яких вони побудовані, були б досить добре підібрані, аксіоматизовані і зрозумілі.

### CRF-модель онтології ПдО

На практиці важливим моментом подання та верифікації онтологій є візуалізація множин ( $D, \mathcal{R}$ ) у вигляді онтологічного графа (онтографа). Останній, в найпростішому випадку, представляється графом у вигляді дерева і описує ієрархічну таксономію понять і відношень між ними. Онтограф має кореневу вершину, якій зазвичай присвоюється ім'я і значення досліджуваної ПдО і "нульовий" рівень ієрархії. Рівні понять, які лежать нижче, формуються у відповідності з прийнятою таксономією. Більш докладні рекомендації щодо структурування онтографів наведені в [12].

У цьому випадку онтологія подається відомою моделлю у вигляді трійки:

$$O = \langle X, R, F \rangle, \quad (1)$$

де  $X$  — множина понять (термінів) на  $D$ ,  $R$  — множина концептуальних відношень, які зв'язують між собою концепти  $X$ ,  $F$  — множина функцій інтерпретації на  $X$  і/або  $R$  (аксіоматизація ПдО і словника на мові деякої

теорії першого порядку). По суті, множини ( $X, R$ ) визначають структуру онтографа. Назвемо модель (1) *CRF (concept–relation–interpretation function)* моделлю.

Для розширеного прикладу будівельної компанії онтограф та індивіди класів представлені на рис.1 (сам онтограф показаний вище потовщеної лінії на рівнях  $0 \div 4$ ). Поняття  $X$  описані на природній мові, розшифрування відношень  $R$  показано на рис. 1 вгорі і праворуч, а функції інтерпретації  $F$  для понять і відношень представляються інтуїтивно зрозумілими. Кореневою вершиною в онтографі є концепт "Будівельна компанія". Їй присвоєно "нульовий" рівень в ієрархії концептів. Людські ресурси компанії позначені концептами "Дирекція", облік співробітників — "Відділ кадрів" і нарахування заробітної плати — "Бухгалтерія". Основна виробнича діяльність компанії позначена концептом "Виробництво". Їм присвоєно "перший" рівень в ієрархії. І так далі. Підкреслимо, що в *CRF*-моделі вершини онтографа з іменами "Директор", "Виконроб", "Майстер" та "Робочий" є поняттями, а конкретні люди на цих посадах уже відносяться до бази даних компанії.

Слід зазначити, що в *CRF*-моделі відношення в онтографі (наприклад,  $R_1$ ) може зв'язувати вершину-предка з декількома вершинами-нащадками.

Передбачається за замовчуванням, що вершина-предок є цілим для частин вершин-нащадків (для відношення  $R_1$  — ціле-частина). Таке твердження вірне для прикладу, який розглядається, поки не визначена конкретна (формальна) мова опису онтологій. В *OWL*-мові відношення (*Object Property*) є бінарними, що зв'язують між собою індивідів двох класів, тому в *OWL*-описах і онторедаторі *Protege* їх необхідно вказувати в явному вигляді, що і продемонстровано на рис. 3.

Нижче онтографа ("потовщеної" лінії) вказані конкретні співробітники компанії на поточний проміжок часу, що в *CRF*-моделі не розглядається і відповідає базі даних, а в *owl*-моделі — значенням даних індивідів.

Сформулюємо визначення онтології з урахуванням моделі (1). Комп'ютерна онтологія ПдО — це:

- ієрархічна структура скінченної множини понять, які описують задану ПдО;
- структура являє собою онтограф, вершинами якого є поняття, а дугами — семантичні (концептуальні) відношення між ними;
- поняття і відношення інтерпретуються відповідно до загальнозначущих функцій інтерпретації, взятих з електронних джерел знань заданої ПдО;
- формально онтограф описується однією з мов опису онтологій;
- функції інтерпретації описані в деякій придатній формальній теорії.

## Проблеми виразності мов опису онтологій

З одного боку, відомі неформальні підходи до мови  $L$ , які допускають тільки визначення термінів з їх специфікацією. З іншого боку, відомі формальні підходи, тобто логікові мови, які дозволяють задавати строго формалізовані логікові теорії. Ці протилежні підходи визначають континуум, введений в [13] і представлений на рис. 2. Чіткої межі, де починається критерій формальності в цьому континуумі, не існує. На практиці права категорія логікових мов (див. рис. 2) зазвичай розглядається як формальна. У цій категорії при виборі мови  $L$  розглядають компроміс між виразністю і ефективністю обчислень.

З правого боку континууму розташовані: логіки вищого порядку, повна логіка першого порядку і модальна логіка. Вони досить виразні, але не завжди допускають повних логікових виведень завдяки своїй нерозв'язності. З протилежного боку континууму розташовані менш строгі підмножини логіки першого порядку, зазвичай допускають розв'язні більш ефективні логікові виведення. Логіки з правого боку можна розділити на дві основні групи.

До першої групи входять мови з сімейства дескрипційних логік ( $DL$ ), які є строгими

підмножинами логіки першого порядку. Друга група виходить з традиції логікового програмування ( $LP$ ) [14], відомим представником якого є  $F$ -логіка. Незважаючи на те, що логікове програмування часто використовує синтаксис, який можна порівняти з логікою першого порядку, воно передбачає іншу інтерпретацію формул.

На відміну від теорії моделей Тарського [15], заснованої на логіках першого порядку і дескрипційних логіках, логікове програмування описує тільки підмножину моделей для суджень про семантичне входження формул. Існують різні способи вибору підмножин моделей, які призводять до різної семантики, — всі вони спрямовані на більш ефективну роботу з великими масивами даних, ніж підходи, засновані на логіці першого порядку. Однією з найбільш помітних відмінностей, що виникають в результаті цього різного стилю логікових моделей, є те, що теорії виразного логікового програмування стають не-монотонними.

## Порівняння відомих логікових теорій опису онтологій

Нижче наведено таблицю порівняльних характеристик найбільш відомих логікових теорій опису онтологій. З таблиці видно, що:

1)  $CRF$ -модель онтології описує концептуальну структуру знань ПдО, яка включає ієрархію понять ( $X$ ,  $R$ ) і визначення на природній мові. Методологія розробки цієї моделі онтології докладно описана в [12]. Слід зазначити, що  $CRF$ -модель знань ПдО не знайшла широкого застосування в додатках користувачів, перш за все через відсутність засобів інтеграції з базами даних і апробованої формальної логікової теорії опису моделі. Однак для  $CRF$ -моделі добре опрацьований підготовчий етап і побудови. Відомий інструментарій автоматизованої побудови множин понять  $X$ , відношень  $R$  і функцій інтерпретації  $F$ ;

2)  $Frame$ -модель онтології ПдО і її  $Protege$ -реалізація найбільш широко застосовується в співтоваристві онтологічного інжинірингу. По-

чинаючи з четвертої версії, *Protege* включає (і може підключити додатково в якості *plug in*) найрізноманітніший інструментарій маніпулювання не тільки компонентами моделі онтології, а й перевірки (різонером) узгодженості розробленої онтології, її візуалізації (*Onto Graph, Onto Viz*), розміщення як на локальному комп'ютері (файл з розширеннями *owl* або *xml*), так і в *Semantic Web* для спільного використання (*Shared using*), причому в повній відповідності з стандартами і рекомендаціями консорціуму *W3C*. Інструментарій *Protege* поширюється з відкритим програмним кодом. Відомий і інший інструмент.

*Top Quadrant's commercial Top Braid Composer* розробки онтологічних додатків, функціо-

нально більш ефективний, ніж *Protege*. Недоліком методології створення як цієї (*Frame*), так і подальшої (*OWL*) моделі онтології є відсутність будь-якої підтримки підготовчого етапу формування онтологій.

Як наслідок, інженер зі знань суб'єктивно формує компоненти моделі, а також вручну створює їх структуру і вводить опис. У кращому випадку — при наявності онторедатора, в гіршому — створенням вручну текстового файлу з великою кількістю *RDF*-трийок;

3) мова *OWL* дозволяє описувати класи (концепти) і надає додаткові можливості в порівнянні з *Protege*. У неї більш багатий набір логікових операторів і вона заснована

Таблиця. Порівняльні характеристики найбільш відомих логікових теорій опису онтологій

Найменуваннякомпонент	Моделі онтології ПДО			<i>Frame (Protege)</i>	
	<i>CRF</i>	<i>Frame (Protege)</i>	<i>OWL (OWL2)</i>		
Об'єкт/суб'єкт (о/с)	Поняття/термін	Клас (концепт)	Клас (концепт)	Концепт (одномісний предикат)	<i>TBox</i> <i>RBox</i>
Зв'язок між о/с		Слот	Властивість (бінарне відношення)	Роль (двомісний предикат), <i>RBox</i>	
Аксіоми опису о/с	Функції інтерпретації	Анотації	Анотації	Набір термінологічних аксіом	
Дані	Семантичне (інтенсіональне) відношення	Екземпляр	Індивід	Індивід	<i>ABox</i>
Зв'язки даних	—	Властивість типу екземпляра	Властивість типу даних/характеристики	Набір тверджень про індивіди	
Значення даних	—	Дані	Дані	Дані	
Машина логікового виведення	—	+	+	+	
Інструментарій для побудови і візуалізації ондографів	—	+	—	—	
Конструктори	—	—	Перетин ( $\cap$ ) Об'єднання ( $\cup$ ) Доповнення ( $\neg$ ) Обмеження ( $\forall, \exists$ )	Перетин ( $\cap$ ) Об'єднання ( $\cup$ ) Доповнення ( $\neg$ ) Обмеження ( $\forall, \exists$ ) та інші	
База знань	—	+	+	+	

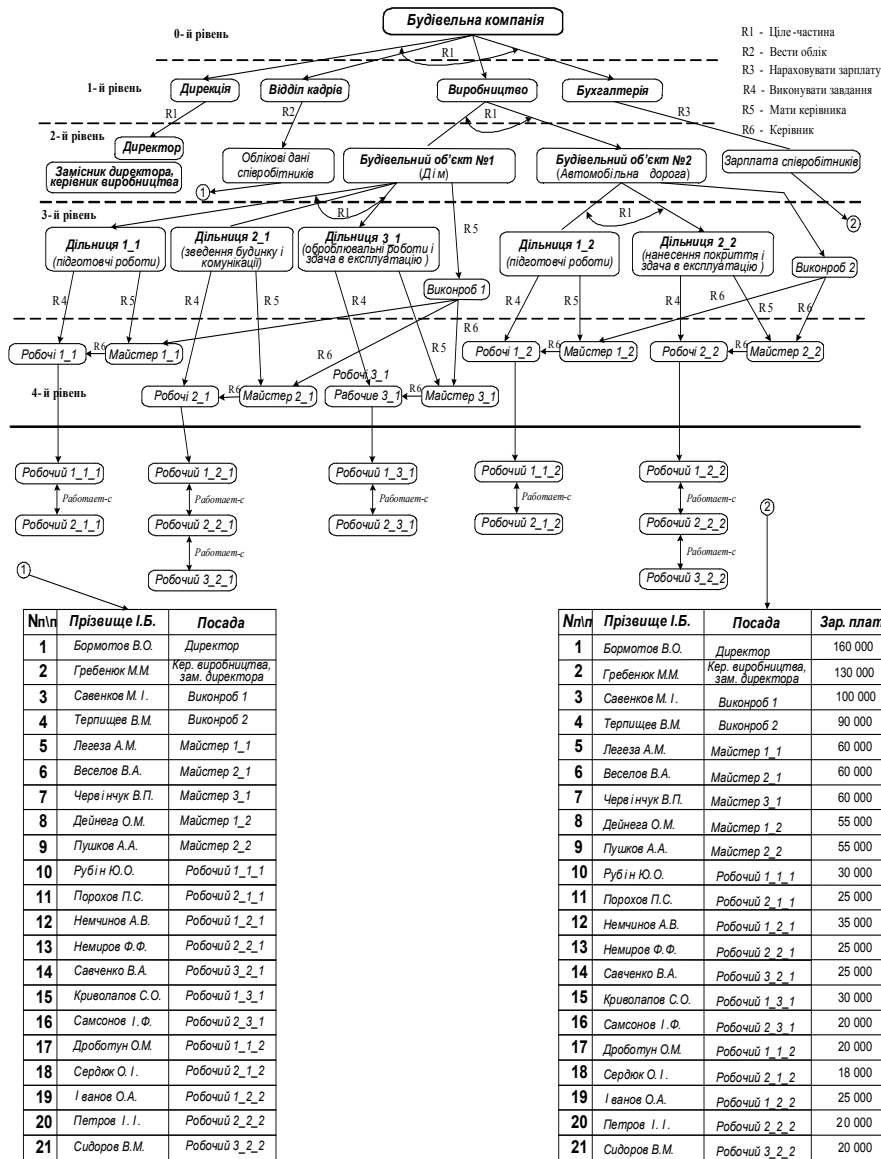


Рис. 1. Онтограф з індивідами класів “Будівельна компанія”

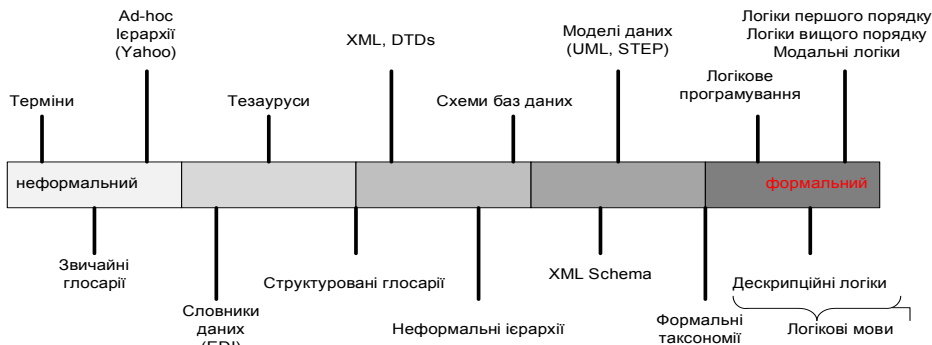


Рис. 2. Різні підходи до вибору мови L



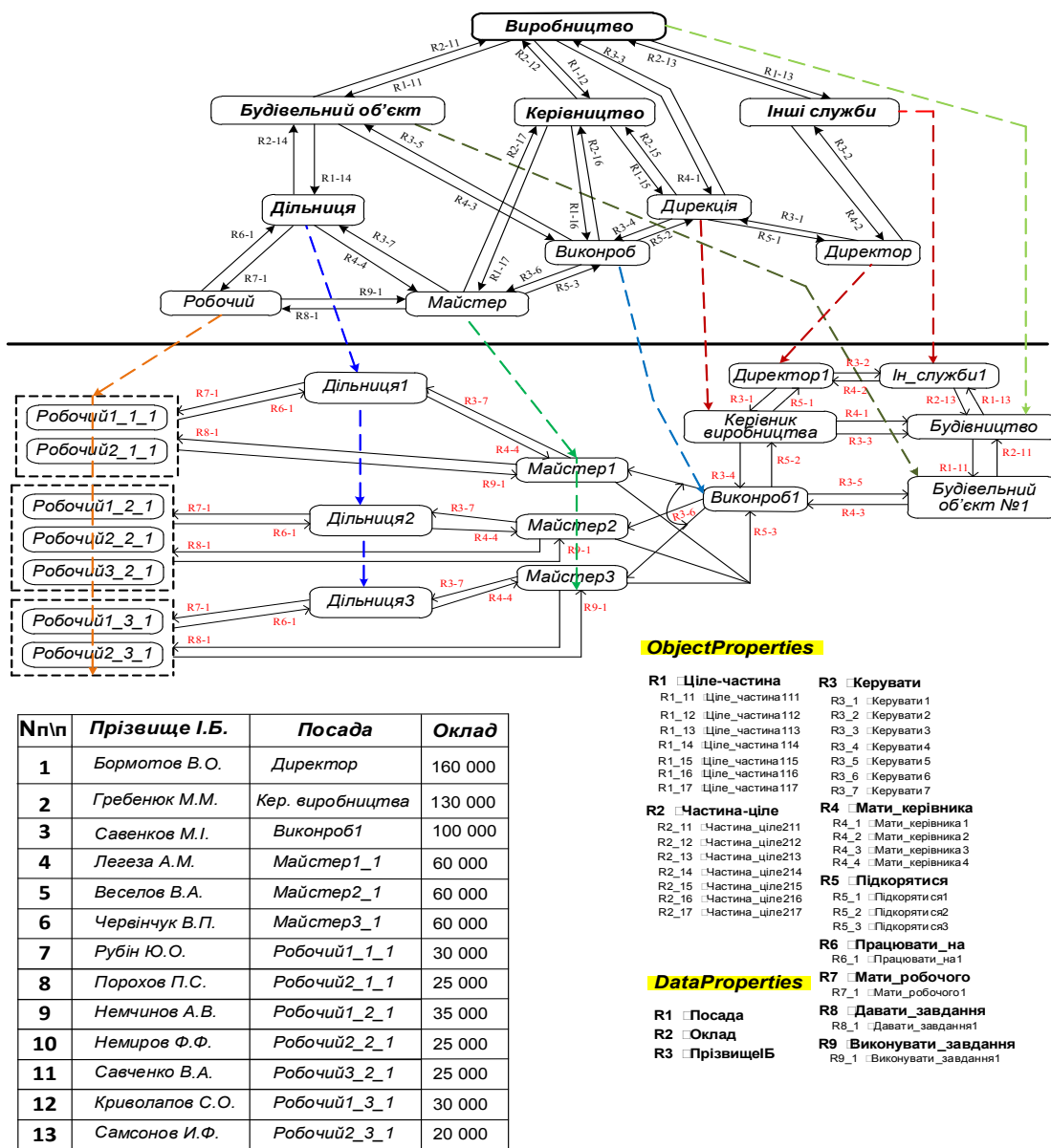


Рис. 3. Protege-сумісний онтограф

на дещо іншій логіковій моделі. Крім того, остання дозволяє використовувати механізм міркувань — різонер, який дозволяє автоматично виводити ієрархію класів і підтримувати її в коректно побудованому стані (перевірка на сумісність і несуперечливість). Починаючи з четвертої версії, в Protege інтегровані в якості *plug in* багато функціональних можливостей мови OWL

і розширених в п'ятій версії. В даний час актуальною є мова OWL2 DL, яка дозволяє описувати онтології як веб-документи і розміщувати їх в WWW для спільного використання з іншими owl-онтологіями [16];

4) останній стовпчик в таблиці дає загальне уявлення про дескрипційну логіку (цілу підмножину логік) і порівняння її компонент з компонентами онтологічних моделей.

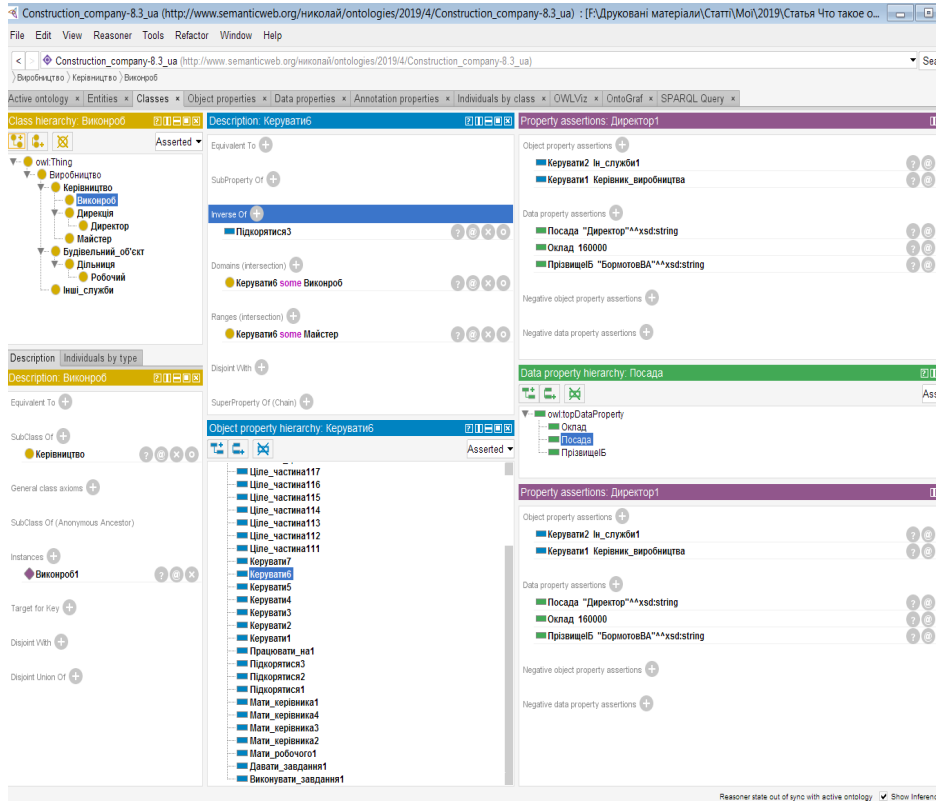


Рис. 4. Приклад представлення основних результатів розробки owl-онтології

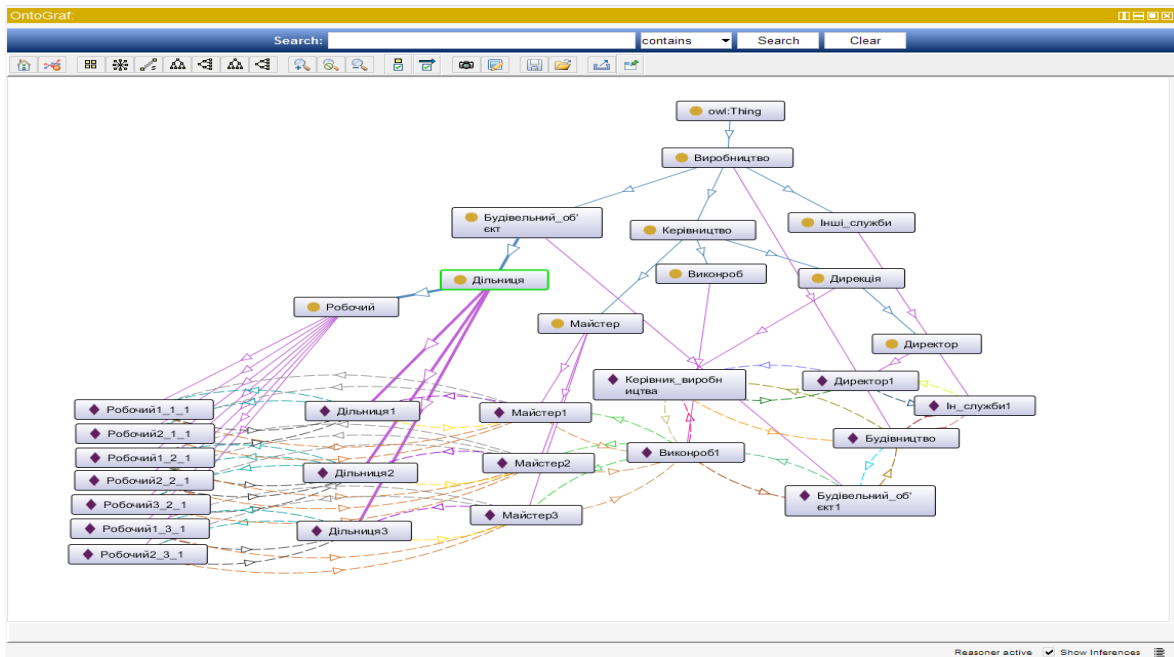


Рис. 5. Онтограф owl-онтології “Будівельна компанія”

## Приклад побудови OWL-онтології “Будівельна компанія”

Формальна онтологія розробляється для прикладу, представленого на рис. 1, в середовищі онторедактора *Protege* 5.5.0 з вбудованими *plugins*-різонером *Pellet*, візуалізацією онтографів *OntoGraf* і механізмом *SPARQL*-запитів. За твердженням авторитетних джерел [16–18] цей онторедактор є таким, що найбільш широко застосовується в співтоваристві онтологічного інжинірингу, в тому числі із-за можливості подання текстових описів в мультимедіумному форматі.

Передбачається, що на комп'ютері створена локальна кінцева точка, в якій зберігаються файли розробленої онтології з розширеннями *\*.owl* і *\*.ttl*. Останній файл використовується *SPARQL*-процесором для обробки запитів (в синтаксисі *Turtle*). Зазначимо, що в прикладі онтограф і індивіди (див. рис. 1) дещо скорочені з метою скорочення обсягу *owl*-опису моделі онтології.

Перед введенням інформації в *Protege* рекомендується побудувати відображення сутностей *CRF*-моделі (див. рис. 1) в *Protege*-сумісний онтограф. Для прикладу таке відображення представлено на рис. 3. Відкриваємо головне меню онторедактора і призначаємо онтології *URI* [http://www.semanticweb.org/ніколай/ontologies/2019/4/Construction\\_company-1](http://www.semanticweb.org/ніколай/ontologies/2019/4/Construction_company-1).

*Tutorials* по створенню *owl*-онтологій в *Protege* і *SPARQL*-запитів представлені в загальнодоступних джерелах, наприклад, в [16–18]. На рис. 4 представлено вікно онторедактора, в якому відображені деякі види вкладок онторедактора, в яких показані результати в процесі розробки *owl*-онтології.

На рис. 5 представлено онтограф *owl*-онтології “Будівельна компанія”, розроблений в додатку “*OntoGraf*”, а на рис. 6 — метрика цієї онтології. Як видно з останньої, створення вручну *owl*-опису сутностей онтології і його налагодження є досить трудомісткою роботою. *Protege*, крім усього іншого, створює *log*-файл, в якому вказані помилки проектування, обчислені різонером.

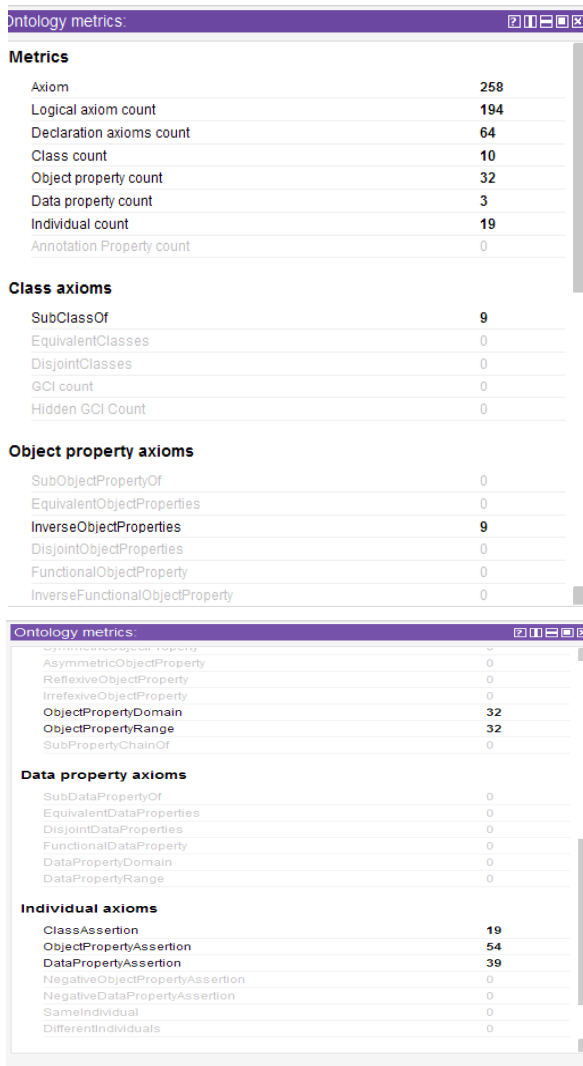


Рис. 6. Метрика OWL-онтології “Будівельна компанія”

Нижче подано деякі *SPARQL*-запити до *owl*-онтології “Будівельна компанія” і відповіді на них.

### Запит №1 — Хто підкоряється Виконробові 1?

```
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
```

```
PREFIX owl: <http://www.w3.org/2002/07/owl#>
```

```
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
```

```
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>
```

```
PREFIX фирма: <http://www.semanticweb.org/ніколай/ontologies/2019/4/Construction_company-8.2#>
```

```
SELECT ?хто
WHERE
{
?хто фірма:Підкоряється 2 фірма:Виконроб 1
}
```

**Відповідь**

Майстер1  
Майстер2  
Майстер3

**Запит№2 — Яке прізвище Робочого 2\_2\_1?**

PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>

PREFIX owl: <http://www.w3.org/2002/07/owl#>

PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>

PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>

PREFIX фірма: <http://www.semanticweb.org/ніколай/ontologies/2019/4/Construction\_company-8.2#>

```
SELECT ?Прізвище
WHERE
{
фірма:Робочий2_2_1 фірма:Прізвище Ім'я По-
батькові ?Прізвище
}
```

**Відповідь**

«Немиров Ф.Ф.»

**Запит№3 — Хто працює на тій же ділянці, що і Немиров?**

PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>

PREFIX owl: <http://www.w3.org/2002/07/owl#>

PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>

PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>

PREFIX фірма: <http://www.semanticweb.org/ніколай/ontologies/2019/4/Construction\_company-8.2#>

```
SELECT ?Робочий ?Прізвище ?Дільниця
WHERE
{
?хто фірма: ПрізвищеІБ «Немиров Ф.Ф.» .
```

```
?хто фірма: Працювати_на1 ?дільниця .
?робочий фірма: Працювати_на1 ?дільниця
?робочий фірма: ПрізвищеІБ ?Прізвище .
}
```

**Відповідь**

Робочий1\_2\_1 «Немчинов А.В.»

Робочий3\_2\_1 «Савченко В.А.»

Дільниця2

**Запит№4 — Скільки отримує Рубін?**

PREFIXrdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>

PREFIX owl: <http://www.w3.org/2002/07/owl#>

PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>

PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>

PREFIX фірма: <http://www.semanticweb.org/ніколай/ontologies/2019/4/Construction\_company-8.2#>

```
SELECT ?Прізвище ?Отримує
WHERE
{
?робочийфірма:ПрізвищеІБ “РубінЮ.О.” .
?робочийфірма:Оклад ?Отримує .
?робочийфірма:ПрізвищеІБ ?Прізвище .
}
```

**Відповідь**

“РубінЮ.О.» «30000»^^<http://www.w3.org/2001/XMLSchema#integer>

**Запит№5 — Хто отримує більше за Рубіна?**

PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>

PREFIX owl: <http://www.w3.org/2002/07/owl#>

PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>

PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>

PREFIX фірма: <http://www.semanticweb.org/ніколай/ontologies/2019/4/Construction\_company-8.2#>

```
SELECT ?Прізвище ?Отримує
WHERE
{
?робочий фірма: ПрізвищеІБ “Рубін Ю.О.”.
```

```
?робочий фірма:Оклад ?оклад Рубіна .
?співробітник фірма:Оклад ?отримує .
? співробітник фірма: ПрізвищеІБ ? прізвище .
FILTER (?отримує> ?оклад Рубіна )
}
```

#### Відповідь

```
«Легеца А.М.» «60000»^^http://www.w3.org/2001/
XMLSchema#integer
«Бормотов В.О.» «160000»^^<http://www.
w3.org/2001/XMLSchema#integer>
«Гребенюк М.М.» «130000»^^<http://www.
w3.org/2001/XMLSchema#integer>
«Савенков М.І.» «100000»^^<http://www.
w3.org/2001/XMLSchema#integer>
«Немчинов А.В.» «35000»^^<http://www.
w3.org/2001/XMLSchema#integer>
«Червинчук В.П.» «60000»^^<http://www.
w3.org/2001/XMLSchema#integer>
«Веселов В.А.» «60000»^^<http://www.w3.org/2001/
XMLSchema#integer>
```

Відзначимо ще одну корисність *CRF*-моделі онтології ПдО.

Згідно [12], вона проектується на основі лінгвістичного корпусу текстів, який описує знання заданої ПдО. З нього добуваються максимально повні (за кардинальністю) множини понять  $X$ , відношень між поняттями  $R$ , а функції інтерпретації  $F$  узгоджуються з визначеннями відповідних понять, наведеними в енциклопедіях, тлумачних словниках та інших наукових публікаціях. Семантика описів по-

нять і відношень між ними максимально погоджена з уявленнями відповідного наукового співтовариства.

*Protege* не перевіряє семантику введених імен класів, властивостей і індивідів. Тобто, семантика онтології, яка розробляється, суб'єктивна, що може викликати проблеми при спільному використанні онтологій.

Отже, при розробці великих за обсягом онтологій, можливо, має сенс заздалегідь розробити *CRF*-модель, що забезпечить семантичну узгодженість такої онтології.

## Висновки

Коротко розглянуто формалізацію визначення онтології предметної області, виконано порівняльний аналіз відомих моделей онтологій, таких як *CRF*, *Frame* и *OWL*. Відзначено їх переваги і недоліки, при цьому зроблено висновок, що при розробці великих онтологій слід застосовувати всі три вказані моделі. На початковому етапі — *CRF*-модель для автоматизації побудови структури концептів предметної області та їх семантичної узгодженості. На етапі введення і формалізації — модель *Protege*-фреймів. І на етапі використання онтології — *OWL*-модель з механізмами запитів та міркувань. При цьому сумісне використання онтології допускає її розміщення в *Semantic Web*. В статті наведено приклад побудови і використання *owl*-онтології для деякої будівельної компанії.

## REFERENCES

1. *Ohrstrom, P., Andersen, J., Schärfe, H.*, 2005. "What has happened to ontology. In F. Dau, M.-L. Mugnier, and G. Stumme, editors", *Conceptual Structures: Common Semantics for Sharing Knowledge*, 13th Int. Conf. on Conceptual Structures, ICCS 2005, Kassel, Germany, July 17–22, 2005, Proceedings, volume 3596 of Lecture Notes in Computer Science, Springer, Heidelberg, pp. 425–438.
2. *Gruber, T.R.*, 1993. A Translation Approach to Portable Ontologies. *Knowledge Acquisition*, 5(2), pp. 199–220.
3. *Gruber, T.R.*, 1995. "Toward Principles for the Design of Ontologies Used for Knowledge Sharing". *International Journal of Human Computer Studies*, 43(5–6), pp. 907–928.
4. *Borst, W.*, 1997. Construction of Engineering Ontologies. PhD thesis, Institute for Telematica and Information Technology, University of Twente, Enschede, The Netherlands.
5. *Studer, R., Benjamins, R., Fensel, D.*, 1998. "Knowledge engineering: Principles and methods". *Data & Knowledge Engineering*, 25(1–2), pp. 161–198.



6. Guarino, N. Giaretta, P., 1995. "Ontologies and Knowledge Bases: Towards a Terminological Clarification". In N. Mars, editor, *Towards Very Large Knowledge Bases: Knowledge Building and Knowledge Sharing*, IOS Press, Amsterdam, pp. 25–32.
7. Genesereth, M.R., Nilsson, N.J., 1987. *Logical Foundations of Artificial Intelligence*. Morgan Kaufmann, Los Altos, CA.
8. Guarino, N., 1998. "Formal Ontology in Information Systems". In N. Guarino, editor, *Formal Ontology in Information Systems*. Proceedings of FOIS'98, Trento, Italy, June 6-8, IOS Press, Amsterdam, pp. 3-15.
9. Carnap, R., 1956. *Meaning and Necessity – A Study in Semantics and Modal Logic*. The University of Chicago Press, second edition.
10. Dowty, D.R. Wall, R., Peters, S., 1980. *Introduction to Montague Semantics*, volume 11 of *Studies in Linguistics and Philosophy*. Springer, Heidelberg.
11. Gruber, T.R., 1993. Towards principles for the design of ontologies used for knowledge sharing. In N. Guarino and R. Poli, editors, *Formal Ontology in Conceptual Analysis and Knowledge Representation*. Kluwer Academic Publishers, Deventer, The Netherlands.
12. Palagin, A.V., Kryvyy, S.L., Petrenko, N.G., 2012. Ontological methods and means of processing subject knowledge. Lugansk: V.I. Dal East Ukr. Nac. University. Retrieved from <http://www.aduis.com.ua/Monography.pdf> (in Russian).
13. Uschold, M., 2004. "Ontologies and Semantics for Seamless Connectivity". *SIGMOD Rec.*, 33(4), pp. 58–64.
14. Das, S.K., 1992. *Deductive Databases and Logic Programming*. Addison Wesley.
15. Vaught, R.L., 1986. "Alfred Tarski's Work in Model Theory". *Journal of Sym. Logic*, 51(4), pp. 869–882.
16. OWL 2 Web Ontology Language Primer, [online] Available at: <http://www.w3.org/TR/2012/REC-owl2-primer-20121211/> [Accessed 16 Oct. 2018].
17. Horridge, M., 2011. *A Practical Guide To Building OWL Ontologies Using Protégé 4 and CO-ODE Tools*. Edition 1.3. Copyright The University Of Manchester, March 24, 107 p.
18. DuCharme, B., 2013. *Learning SPARQL. Querying and Updating with SPARQL 1.1*. O'Reilly Media. All rights reserved, 367 p.

Received 11.07.2019

O.V. Palagin, Academician of NAS of Ukraine, Professor, Doctor of Eng.,  
deputy director of V.M. Glushkov Institute of Cybernetics of NAS of Ukraine,  
Glushkov ave., 40, Kyiv, 03187, Ukraine,  
[palagin\\_a@ukr.net](mailto:palagin_a@ukr.net)

M.G. Petrenko, Doctor of Eng., Leading Researcher,  
V.M. Glushkov Institute of Cybernetics of NAS of Ukraine,  
Glushkov ave., 40, Kyiv, 03187, Ukraine,  
[petrng@ukr.net](mailto:petrng@ukr.net)

## SOME FEATURES OF THE CONSTRUCTION OF ONTOLOGICAL DOMAIN MODELS

**Introduction.** The paper discusses the features of building ontological models of subject areas. A comparative analysis of the most well-known logical theories of the formalization of these models and the corresponding tools for the construction and use of ontologies is performed. An example of using the developed information technology based on the standards and recommendations of the W3C consortium is given.

**Purpose.** Consideration of the most well-known tools for the formal description of domain ontologies, their comparative analysis and the choice of a formal language and ontoeditor for practical use.

**Methods.** The methods and models used in the work are based on the information technologies of the Semantic Web, focused on the development and use of domain ontologies. Ontologies are the basic components of these technologies for both research and the creation of large projects, including commercial ones. The use of methods of system-ontological analysis revealed the advantages and disadvantages of various formal and informal descriptions of ontologies. Of the informal models, the CRF model was considered, and the formal models, the Frame model and the OWL model. It also compares the characteristics of these models with a set of descriptive logics.

**Results.** Information technology has been developed to enable the construction of ontologies of subject areas, including the outline design stage, the input information input into the computer, the formal description of the project, and the practical use of ontologies.

**Conclusion.** The formalization of the definition of the domain ontology is briefly considered; a comparative analysis of known models of ontologies, such as *CRF*, *Frame* and *OWL*, is performed. Their advantages and disadvantages are noted, and it was concluded that when developing large ontologies, all three indicated models should be used. At the initial stage — a *CRF*-model for automating the construction of the structure of domain concepts and their semantic consistency. At the stage of input and formalization — a model of *Protege*-frames. And at the stage of ontology use — the *OWL*-model with the mechanisms of inquiries and reasoning. At the same time, ontology sharing implies its placement on the Semantic Web. The article gives an example of building and using owl-ontology for a certain construction company.

**Keywords:** *descriptive logic, ontology, ontology model, conceptualization, ontograph, Protege.*

*А.В. Палагин*, академик НАНУ, доктор технических наук, профессор, зам. директора, Институт кибернетики им. В.М. Глушкова НАН Украины, просп. Академика Глушкова, 40, Киев, 03187, Украина, palagin\_a@ukr.net

*Н.Г. Петренко*, доктор технических наук, ведущий научный сотрудник, Институт кибернетики им. В.М. Глушкова НАН Украины, просп. Академика Глушкова, 40, Киев, 03187, Украина, petrng@ukr.net

## О НЕКОТОРЫХ ОСОБЕННОСТЯХ ПОСТРОЕНИЯ ОНТОЛОГИЧЕСКИХ МОДЕЛЕЙ ПРЕДМЕТНЫХ ОБЛАСТЕЙ

**Введение.** В статье рассмотрены особенности построения онтологических моделей предметных областей. Проведен сравнительный анализ наиболее известных логических теорий формализации указанных моделей, соответствующего инструментария построения и использования онтологий. Приведен пример использования разработанной информационной технологии, базирующейся на стандартах и рекомендациях консорциума *W3C*.

**Цель статьи** — анализ наиболее известных инструментов формального описания онтологий предметных областей, их сравнительный анализ и выбор формального языка онторедатора для практического использования.

**Методы.** Методы и модели, используемые в работе, базируются на информационных технологиях *Semantic Web*, ориентированных на разработку и использование онтологий предметных областей. Онтологии есть базовыми компонентами указанных технологий как для проведения научных исследований, так и для создания емких проектов, в том числе коммерческого характера. Использование метода системно-онтологического анализа позволило определить преимущества и недостатки разных формальных и неформальных описаний онтологий. Из неформальных моделей была рассмотрена *CRF*-модель, а из формальных — *Frame*-модель и *OWL*-модель. Также проведено сравнение характеристик указанных моделей с множеством дескрипционных логик.

**Результаты.** Разработана информационная технология, позволяющая реализовать этапы построения онтологий предметных областей, включая этап эскизного проектирования, введения в компьютер входной информации, этап формального описания проекта и этап практического использования онтологий.

**Выводы.** Коротко рассмотрено формализацию определения онтологии предметной области, выполнен сравнительный анализ известных моделей онтологий, таких как *CRF*, *Frame* и *OWL*. Определены преимущества и недостатки, при этом сделан вывод, что при разработке больших онтологий необходимо использовать все три указанные модели. На начальном этапе — *CRF*-модель для автоматизации построения структуры концептов предметной области и их семантической согласованности. На этапе введения и формализации — модель *Protege*-фреймов. И на этапе использования онтологии — *OWL*-модель с механизмами запросов и рассуждений. При этом совместное использование онтологии допускает ее размещение в *Semantic Web*. В статье приведен пример построения и использования owl-онтологии для некоторой строительной компании.

**Ключевые слова:** *дескрипционная логика, онтология, модель онтологии, концептуализация, онтограф, Protege.*