

## ВИКОРИСТАННЯ ОНТОЛОГІЧНИХ ЗНАНЬ ІНТЕЛЕКТУАЛЬНИМИ АГЕНТАМИ

Розглянуто онтологічні засоби представлення знань в інтелектуальних інформаційних системах, орієнтованих на роботу з Web. Запропоновано формальну модель онтології, що забезпечує використання знань з інших розподілених застосувань, які підтримують семантичну обробку інформації. Специфіка цієї моделі пов'язана з тим, що в ній виокремлюються такі властивості онтологічних елементів, як ієрархія, перетин та еквівалентність. Розглянуто, як відрізняється обробка онтологій в інтелектуальних інформаційних системах відповідно до їх ролі – внутрішніх та зовнішніх. Проаналізовано властивості таких онтологій та їх використання персональними програмними агентами, поведінка яких моделюється через апарат інтенціональних відношень (переконання, знання та цілі). Персональні програмні агенти розглядаються як підклас комп'ютерних агентів. Проаналізовано рівень інтелектуальності таких агентів та їх здатність керування розподіленими знаннями.

Ключові слова: онтологія, інтелектуальний програмний агент, Semantic Web, онтологічна модель, інтенціональні відношення.

### Вступ

Сучасні *інтелектуальні інформаційні системи* (ІС), що працюють у відкритому розподіленому інформаційному просторі, потребують постійного поповнення і відновлення знань, що надходять із зовнішнього середовища, тому що в ІС використання знань про *предметну область* (PrO) і методи їхньої обробки відіграють вирішальну роль і визначають корисність отриманих результатів.

У зв'язку з тим, що сучасні ІС використовують і генерують великі за обсягом і складні за структурою сукупності знань, виникає проблема розробки ефективних методів керування ними.

Використання розподілених знань, які ІС отримують з Web, є одним з визначальних факторів їх ефективності. Це у свою чергу обумовлює актуальність досліджень, спрямованих на отримання і накопичення знань, рішення проблем розпізнавання, логічного виведення нових знань для підтримки прийняття рішень, тобто на розвиток засобів керування знаннями, в тому числі – онтологічними.

### Онтологічне подання знань

В даний час для інтероперабельного представлення розподілених знань у ІС широко застосовуються онтології, що забезпечують повторне використання знань [1]. За останні роки онтології активно роз-

виваються як семантичні моделі, що дозволяють відображати різні аспекти реального світу [2].

Оскільки онтологія явно представляють собою предметні області через існуючі сутності, їх властивості та відношення між ними, вони можуть використовуватися для підтримки гетерогенно структурованих баз даних та знань і різноманітних ІС систем, забезпечуючи співставлення їх семантики. Онтології, таким чином, підтримують семантичну інтероперабельність і інтеграцію в організаціях у різних предметних областях.

Однак, існують великі відмінності між цими онтологіями. Потенційні користувачі цих онтологій зазвичай не мають ніякого уявлення щодо їх сфери охоплення, їх зрозумілості для людини, їх достовірності та надійності, їх стійкості, типів логічного виведення, для яких вони можуть бути використані, або їх адаптивності і здатності використовуватися повторно для інших цілей.

Користувачам важко зрозуміти, чи можуть конкретні онтології допомогти їм вирішити їхні конкретні проблеми. Їм не відомо, як саме великі онтології утворюються із злиття або відображення менших онтологій. Розвиток онтологій потребує більш строгого підходу до їх розробки і порівняння.

Онтології базуються на дескриптивних логіках, для них вже існують загальноприйняті стандарти опису, мови і програмні засоби. Нині для подання онтологій найчастіше використовують різні діалекти OWL та RDF, які відрізняються своїми виразними можливостями і своєю складністю: RDF Schemas пропонує найпростіший рівень для представлення онтологій, а OWL Full – найбільш складний. Вибір засобу представлення онтології залежить від специфіки проблеми, для якої вона розробляється [3].

У різних джерелах пропонуються відмінні між собою формальні моделі подання онтологій, що дають змогу адекватно та ефективно моделювати уявлення про різноманітні ПрО та формально задавати їх семантику [4], які відрізняються як основними параметрами, так і виразною здатністю, що залежать від цілей розробки таких моделей.

Онтології містять класи та екземпляри класів, їх властивості, відношення та обмеження використання. Фундаментальні поняття мають відповідати класам онтології. Для визначення екземпляра досить оголосити його членом якого-небудь класу. *Властивості* дають можливість затверджувати загальні факти про класи і про екземпляри. При визначенні властивості існує багато способів обмежити це відношення, приміром, задати його домен і діапазон або визначити як спеціалізацію (підвластивість) вже існуючої властивості. Можливі і більш складні обмеження. Властивості, також як і класи, можуть бути організовані ієрархічно.

Надалі будемо використовувати таку формальну модель:

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

Ця модель складається з таких елементів:

–  $X = X_{cl} \cup X_{ind}$  – множина концептів онтології, де  $X_{cl}$  – множина класів,  $X_{ind}$  – множина екземплярів класів, таких, що  $\forall a \in X_{ind} \exists A \in X_{cl}, a \in A$ ;

–  $R = r_{ier\_cl} \cup \{r_i\} \cup \{p_j\}$  – множина відношень між елементами онтології,

де  $r_{ier\_cl}$  – ієрархічне відношення, що може встановлюватися між класами онтології і властивостями класів і характеризується такими властивостями, як антисиметричність і транзитивність,  $r_{ier\_cl} : X_{cl} \rightarrow \rightarrow X_{cl}; \{r_i\}$  – множина об'єктних властивостей, що устанавлюють відношення між екземплярами класів:

$$r_i(a, a \in X_{ind}) = b, b \in X_{ind},$$

$$r_i : X_{ind} \rightarrow X_{ind};$$

$\{p_j\}$  – множина властивостей даних, що встановлюють відношення між екземплярами класів і значеннями:

$$p_i(a, a \in X_{ind}) = t, t \in T, p_i : X_{ind} \rightarrow Const,$$

такі, що усередині множин об'єктних властивостей і властивостей відношень також можуть існувати ієрархічні відношення  $r_{ier\_obj}, r_{ier\_obj} : \{r_i\} \rightarrow \{r_i\}$  і  $r_{ier\_data}, r_{ier\_data} : \{p_j\} \rightarrow \{p_j\}$ ;

–  $F$  – множина характеристик класів онтології, екземплярів класів і їхніх властивостей, що можуть застосовуватися для логічного виведення (наприклад, еквівалентність, відмінність, відсутність перетину, область визначення та область значення);

–  $T$  – множина типів даних (наприклад, рядок, ціле).

Вибір саме такої моделі онтології для даної роботи обумовлюється наступними причинами. По-перше, така модель має достатню виразність для розв'язання тих задач, які розглядаються в цій роботі, а саме, керування розподіленими знаннями в МПІС. По-друге, вона відповідає інтуїтивному уявленню про онтології, яке міститься в користувальницькому інтерфейсі редактора онтологій Protégé [5] і тому легко поєднується з представленням елементів онтології в цьому програмному продукті. По-третє, ця модель досить просто інтегрується з різними інтелектуальними застосуваннями, які підтримують семантичну обробку інформації (приміром, з семантичними Wiki-ресурсами, лексичними онтологіями).

## Внутрішні та зовнішні онтології

Як правило, у ІС розрізняють внутрішні і зовнішні онтології [6]. Внутрішні онтології створюються розроблювачами ІС і відображають їхнє уявлення щодо властивостей та взаємовідношень основних *інформаційних об'єктів* (ІО) певної ПрО. Структура внутрішньої онтології і її обсяг цілком визначаються цілями створення ІС і тими функціями, що вона має виконувати. Структура внутрішньої онтології залишається незмінною у процесі функціонування ІС і може лише поповнюватися новими екземплярами класів або значеннями їх властивостей.

Зовнішні онтології імпортуються ІС із зовнішніх інформаційних джерел у процесі роботи системи і дозволяють динамічно поновлювати відомості щодо відповідної ПрО. Приміром, такі онтології можуть бути знайдені за певними умовами в репозиторіях онтологій або бути побудовані шляхом аналізу доступних *інформаційних ресурсів* (ІР). Важливою властивістю зовнішніх онтологій є те, що в процесі роботи ІС для рішення тієї самої задачі можуть застосовуватися різні онтології. Це пояснюється динамічністю інформаційного середовища ІС (це особливо характерно для ІС, орієнтованих на роботу в Web): онтології у репозиторії або зовнішні ІР можуть значно змінюватися незалежно від розробників та користувачів ІС.

Такі властивості Web, як гетерогенність і динамічність, обумовлюють ряд проблем, пов'язаних з використанням зовнішніх онтологій. Зокрема, виникає проблема порівняння зовнішніх онтологій, сформованих у різні моменти роботи ІС: якщо відмінності між онтологіями перевищують певну кількісну міру, тоді необхідно заново виконати їхню обробку.

Зважаючи на те, що ресурси, за якими будуються онтології, та методи, що використовуються для цього, заною мірою визначають властивості та структуру онтологій, доцільно розглянути окремі випадки таких онтологій. Якщо зовнішню онтологію можна віднести до одного з таких типів, це значно спрощує її обробку та використання.

*Таксономія*  $T$  – це онтологія, яка, крім класів та екземплярів, містить тільки відношення “підклас”, яке ієрархічно поєднує класи, та відношення “екземпляр класу”. Ця онтологія не потребує складних засобів обробки, методи її побудови не припускають наявності циклів у класах (кожен клас створюється як підклас певного класу, екземпляр – як екземпляр певного класу). Її формальна модель є окремим випадком моделі (1):

$$T = \langle X = X_{cl} \cup X_{ind}, R = \{r_{ier-cl}\}, \emptyset, \emptyset \rangle. \quad (2)$$

*Wiki-онтологія* – це онтологія, побудована за семантично розміченим Wiki-ресурсом (набором Wiki сторінок, що містять семантичну розмітку). Вона містить тільки ті знання, які можна безпосередньо здобути із семантичної розмітки. Тому в цій онтології відсутні, приміром, такі характеристики класів та властивостей, як еквівалентність, відсутність перетину тощо. Але ця онтологія може містити суперечності, пов'язані з класифікацією екземплярів, які потребують додаткової перевірки [7].

Її формальна модель також є окремим випадком моделі (1):

$$T_{wiki} = \langle X, R, \emptyset, T \rangle. \quad (3)$$

В цій моделі множина концептів будується як поєднання таких елементів Wiki, як сторінки та категорії

$$X = X_{wiki\_categor} \cup X_{wiki\_page},$$

пов'язаних різними видами відношень з

$$R = \{r_{ier-cl}\} \cup \{r_{link}\} \cup \{r_{sem\_prop}\},$$

тобто множина класів – це множина категорій Wiki

$$X_{wiki\_categor},$$

між якими існують ієрархічні відношення  $r_{ier-cl}$ ; множина екземплярів – це множина Wiki-сторінок  $X_{wiki\_page}$ , між якими існують посилання  $r_{link}$  та семантичні відношення  $r_{sem\_prop_i}, i = \overline{0, m}$ ; а множина типів даних доповнюється специфічним класом – “Wiki-сторінка”. Ця модель може

бути вдосконалена з урахуванням таких елементів Wiki, як шаблони, форми, спеціальні сторінки тощо.

### Агентний підхід до моделювання ІС

Сьогодні у розробці складних ІС, орієнтованих на роботу в Web, досить широко застосовується агентно-орієнтоване програмування (АОП). Ця парадигма програмування дає змогу перейти на більш інтелектуальний рівень взаємодії користувача з програмним і апаратним забезпеченням [8]. Можна розглядати агентний підхід як метафору проектування та моделювання розподілених систем.

*Програмні агенти* (ПА) – це програмні сутності, які функціонують безперервно й автономно в конкретному оточенні здебільшого разом з іншими процесами й агентами. Можна розглядати ПА як окремих випадок автономних комп'ютерних агентів [9]. Значна частина сучасних ПА та *мультиагентних систем* (МАС) орієнтовані саме на автономне функціонування у відкритому інформаційному середовищі Web, враховують його специфіку та здатні використовувати його IP. Підвищення ефективності використання ПА пов'язане з їх інтелектуалізацією. Тому великий інтерес викликає проблема керування розподіленими знаннями в *мультиагентних ІС* (МІС).

Розробка агентно-орієнтованого програмного забезпечення розширює можливість інформаційних технологій, забезпечуючи більш гнучку композицію ІС з менш тісно пов'язаними компонентами, спрощує проектування і зменшує час впровадження. Але методи, що ефективно використовуються у традиційних ІС, можуть бути застосовані до створення агентів та мультиагентних систем.

Для ПА характерна *раціональність* поведінки, тобто ПА виконують тільки такі дії, які, відповідно до обізнаності агента, мають привести до досягнення поставлених користувачем цілей. Певну програмну сутність доцільно розглядати саме як ПА, якщо вона має хоча б деякі з наступних властивостей:

– *автономність* – здатність самостійно виконувати розпорядження користувача, використовуючи знання про користувача та досвід взаємодії з ним;

– *реактивність* – можливість змінювати свою поведінку відповідно до зміни ситуації;

– *комунікабельність* – здатність ПА до обміну інформацією з іншими ПА та програмними компонентами;

– *коллаборативність* – здатність співпрацювати з іншими ПА.

Більш детально підходи до класифікації ПА проаналізовано в [10].

Через складність та неоднозначність терміну “ПА”, досить складно сформулювати відмінність агентів від інших типів програмного забезпечення. Досить перспективним уявляється підхід, запропонований в [11], де виділяються шість основних характеристик різних видів агентів: інформованість (knowledgeable), співпраця (collaboration), адаптивність (adaptability), стійкість (persistence), мобільність (mobility) та автономність (autonomy). Кожна програмна сутність має певну кількісну оцінку кожного з цих параметрів та займає певну позицію у такому шестивимірному просторі. Ці шість ортогональних характеристик працюють разом, щоб агент-орієнтовані системи мали більшу гнучкість. Чим більшу площу має вона на діаграмі, тим більш доцільно розглядати відповідну програму як ПА.

Для класифікації ПА, залежно від її цілей, застосовують такі параметри, як рівень інтелектуальності, призначення, платформа виконання тощо. Для *інтелектуальних* програмних агентів (ІПА) характерні *спілкування на рівні знань* – здатність у взаємодії з користувачами та іншими ПА, використовуючи зовнішні БЗ, здатність до *логічного виведення* та *адаптивності* – здатність навчатися та вдосконалювати свою поведінку на основі власного досвіду. Через відмінності агентів від інших типів програмного забезпечення досить важко чітко схарактеризувати їх інтелектуальність, що визначається їх здатністю до міркувань і навчання [12]. Вищий рівень інтелектуальності припус-

кає наявність в агента моделі користувача, його потреб і механізму пошуку засобів їх задоволення. Найбільш інтелектуальні системи здатні до навчання на власному досвіді й адаптації до динамічного середовища.

Можливість навчання є важливою властивістю будь-якої системи, яка функціонує у середовищі, що динамічно змінюється. У МПС проблеми навчання особливо актуальні, але мають ряд особливостей. При цьому інтелектуальність розглядається вже не як властивість окремого ПА, а як властивість групи взаємодіючих агентів [13]. Кожен ППА в МПС може мати власне уявлення щодо свого середовища. Це уявлення може бути формалізовано через відповідну онтологію [14].

### Персональні агенти – агенти користувачького інтерфейсу

З точки зору моделювання МПС найбільшу цікавість викликають ті агенти, які безпосередньо взаємодіють з користувачем, на відміну від інших ПА, що отримують користувачькі завдання опосередковано, від інших ПА. Це викликано тим, що саме їх дії користувачі можуть спостерігати безпосередньо і тому прогнозування та пояснення цих дій найбільш цікавить користувачів.

*Персональні програмні агенти* (ППА) – це ПА, які мають засоби взаємодії з користувачем (інтерфейс) для того, щоб отримувати від нього певні завдання та передавати результати їх виконання, виступаючи в ролі посередників для взаємодії з усіма іншими агентами [11, 15].

Дії ППА залежать не лише від завдання користувача, але й від того, який саме користувач дає це завдання, тобто ППА здатні враховувати інформацію щодо користувача та щодо досвіду взаємодії з ним для того, щоб покращити свою роботу. Для цього вони використовують модель користувача, хоча б і неявну, яка дозволяє прогнозувати дії користувача, його потреби та реакцію. Така неявна модель користувача має взаємодіяти з моделями задачі і ПрО. У більш складних інтерфейсних агентів корисно робити мо-

дель користувача більш явним, так щоб вона дозволяла здійснювати логічне виведення. Крім того, доцільно перейти від статичних моделей до динамічних, де можна відстежувати функції, якими користувач маніпулює.

В літературі ППА також називають агентами інтерфейсу або користувачькими агентами.

ППА, спілкуючись з людиною безпосередньо через вхід і вихід інтерфейсу, спостерігає за її діями та за об'єктами, якими ця людина маніпулює, має доступ до сенсорів та ефекторів і може сам виконувати певні дії [14].

ППА можуть виконувати завдання користувачів, що стосуються автоматизації обробки даних та прийняття певних рішень. В роботі [16] виділяють наступні підтипи інтерфейсних агентів:

- агенти інтерфейсу з традиційними застосуваннями: людина взаємодіє з програмою, як звичайно, а агент спостерігає за її діями та пропонує певні рішення, щоб запобігти можливих труднощів;

- традиційний інтерфейс для мультиагентних систем: людина безпосередньо взаємодіє з агентами МАС через традиційний інтерфейс;

- агенти інтерфейсу для МАС: інтеграція людини з МАС, де людина розглядається як компонента системи (агент), в якій вона грає одну або кілька ролей.

За ролями ППА можна поділяти на помічників і радників. Вони забезпечують адаптований, персоніфікований інтерфейс, враховуючи індивідуальні особливості, унікальні потреби і переваги користувача.

Для ефективного використання ППА потрібно, щоб користувач відчував довіру до нього, тобто міг прогнозувати його дії, тому що делегування ППА певних функцій пов'язане з певним ризиком того, що агент неправильно зрозуміє наміри користувача або, що це завдання може бути виконано неправильно. Для цього важливо забезпечити зворотний зв'язок з агентом та надати користувачеві можливість впливати на його роботу. Тому потрібно, щоб користувач розумів фу-

нкції окремих компонентів МПС та вплив своїх дій на результати роботи.

Інтелектуальні ППА, що входять до складу МПС або діють самостійно, здатні використовувати формально представлені знання щодо користувача і тієї ПрО, що його цікавить, здобувати ці знання з різних джерел та здійснювати логічне виведення на цих знаннях.

### BDI-модель агента

Для того, щоб моделювати МПС та мати можливість прогнозувати її дії, доцільно використовувати *інтенціональні відношення* (ІнВ) [17], які зазвичай використовуються для прогнозування й пояснення людської поведінки. ІнВ поділяються на інформаційні відношення (переконавання, знання) і перед-відношення (бажання, емоції, наміри, зобов'язання, звершення, вибір, цілі тощо). Інформаційні ІнВ пов'язані з відомостями ППА про світ, у якому він існує, тоді як перед-відношення – з діями ППА. Модель ПрО складається з переконавань і знань агента про її концепти та зв'язки між ними. Цей підхід застосовують тоді, коли інформації про властивості системи та її структуру бракує, або вона надто складна для аналізу.

Якщо ППА використовують онтологічне подання знань, то важливо зв'язати ці інтенціональні відношення з елементами онтологічної моделі.

*Переконавання* (beliefs) ППА відображають його уявлення про поточний стан світу та про зміни в ньому, які можуть статися внаслідок виконання дій ІА.

Переконавання ППА поділяються на:

- *внутрішні*, які ППА отримує при розробці;
- *набуті*, які з'являються в ППА в процесі його функціонування внаслідок спостережень і оцінки інформації.

Переконавання референційно непрозорі. Перед- та інформаційні відношення тісно пов'язані, оскільки агенти можуть, наприклад, формувати наміри на основі наявної в них інформації про світ. Модель ППА, у якій використовуються ІнВ, звичайно, називають *BDI-моделлю* (від

Believe, Desire, Intention), навіть якщо ця модель використовує інший набір ІнВ. Для формалізації ІнВ та моделювання поведінки ППА використовується апарат модальної логіки.

*Бажання* (desire) відображають ставлення ІА до майбутніх станів світу та переваги, які він надає одним з них над іншими. Важливою рисою бажання є те, що ІА може мати несумісні й недосяжні бажання.

*Цілі* (goals) – це несуперечлива підмножина бажань ІА.

*Наміри* (intentions) – це підмножина цілей, яких може досягти обмежений у ресурсах ППА, і засіб їх досягнення.

За допомогою інтенціональних відношень відображаються компоненти моделі інтелектуального агента. Модель предметної області складається з переконавань і знань агента про її концепти та зв'язки між ними. Модель користувача містить переконавання агента, пов'язані зі взаємодією з користувачем. Моделі інших агентів дають змогу агенту використовувати для досягнення своїх намірів і цілей знання та переконавання цих агентів тощо.

Аналіз агентів на рівні знань, який запропонував Невелл [18], передбачає, що агенти мають деякі знання й цілі та вважають, що можуть досягти цих цілей, використовуючи наявні в них знання. Поведінку агентів на рівні знань можна описати за допомогою принципу раціональності: якщо агент має знання про те, що одна з його дій приведе до однієї з його цілей, то агент вибере цю дію.

Модель ППА, у якій використовуються ІнВ (не обов'язково саме переконавання, бажання та наміри), зазвичай називають *BDI-моделлю* (Belief-Desire-Intention). Наведений вище аналіз публікацій свідчить про великий інтерес до використання різних комбінацій ІнВ для моделювання поведінки ІА, вибір якої визначається специфікою інтересів розробників моделі: Кохен і Левеск'є використовують як базові відношення переконавання та цілі (інші відношення визначаються через них) [19], Рао і Джорджеф [20] – переконавання, бажання та наміри, Братман та

ін. [21] – більш практичні ІнВ – цілі та плани.

Ми пропонуємо формалізацію чотирьох ІнВ – знань, переконань, намірів та цілей [22]. Для цього використовується апарат модальної логіки, модальні оператори якої відповідають основним характеристикам ППА, які потрібно відобразити в його символічній моделі. Такий формалізм може використовуватися для опису поведінки різних ППА і МПС.

Модальна логіка, яка формалізує ці ІнВ (BDI-логіка), може використовуватися для моделювання поведінки ППА, які мають здатність до цілепокладання та власну модель світу, що будується на основі знань і переконань.

Розглянемо твердження типу «А (R) р», де А – суб’єкт, р – твердження, а R – відношення між А і р. Якщо як приклад відношення R розглянути переконання, то твердження, що «А вважає р» істинним, означає, що р визнається істинним для А, незалежно від справжньої істинності р, тобто є два погляди: А – внутрішнього і К – зовнішнього суб’єктів. Зафіксуємо базисну множину W, яка віддзеркалює істинну картину й відповідає знанням зовнішнього суб’єкта К. У W оцінюються як немодальні, так і модальні висловлювання. U – допоміжна множина можливих світів, яка презентує знання внутрішнього суб’єкта А. У світах  $u \in U$  оцінюються тільки немодальні висловлювання, які виражають судження внутрішнього суб’єкта про об’єкти. Світ  $u \in U$  відображає світ  $\varpi \in W$  з погляду А.

Як загальнозначущий критерій відмінності знання від переконання береться фіксований погляд, репрезентований базовою системою можливих світів W. У такому разі «А вважає р» істинним у світі  $\varpi \in W$  тоді й тільки тоді, коли р істинне в співвіднесеному зі світом  $\varpi \in W$  світі  $u \in U$ . «А знає р» істинно в  $\omega$  тоді й тільки тоді, коли р істинно у світі  $\omega$  і в співвіднесеному з  $\omega$  світі  $u \in U$  (суб’єкт підтверджує істинність р, і р справді істинне).  $\omega$  – класичний можливий світ (у ньому виконуються закони класичної логіки й він цілком визначений: будь-яке пра-

вильно побудоване висловлювання в ньому або істинне, або хибне). Але суб’єкт А не завжди має у своєму розпорядженні повну й точну інформацію про світ, тому його уявлення про світ складаються з здогадів і припущень. Через те для опису світів з U двох значень «істинно» та «хибно» недостатньо (наприклад, є твердження, істинна оцінка яких невідома, але суб’єкт упевнений у можливості її отримання).

$\mathfrak{S}$  – мова класичної пропозиційної логіки.  $\mathfrak{S}^e$  – епістемічна мова, що містить множину пропозиційних змінних  $\text{Var} = \{p, q, r, \dots\}$ , зв’язки & і , дужки і сімейство узагальнених модальних операторів

$$\{B_i, K_i, D_i, I_i: i = \overline{1, N}\}.$$

F – множина формул мови  $\mathfrak{S}$ , де:

- якщо  $\alpha \in \text{Var}$ , то  $\alpha \in F$ ;
- якщо  $\alpha \in F$ , то  $\neg\alpha \in F$ ;
- якщо  $\alpha, \beta \in F$ , то  $(\alpha \& \beta) \in F$ .

Множина формул мови  $\mathfrak{S}^e$  – така множина  $F^e$ , що:

- якщо  $\alpha \in F$ , то  $\alpha \in F^e$ ;
- якщо  $\alpha \in F^e$ , то  $B_i\alpha \in F^e$ ,  $i = \overline{1, N}$ ;
- якщо  $\alpha \in F^e$ , то  $D_i\alpha \in F^e$ ,  $i = \overline{1, N}$ ;
- якщо  $\alpha \in F^e$ , то  $I_i\alpha \in F^e$ ,  $i = \overline{1, N}$ ;
- якщо  $\alpha \in F^e$ , то  $K_i\alpha \in F^e$ ,  $i = \overline{1, N}$ .

Інші зв’язки ( $\vee, \supset, \equiv$ ) визначаються звичайним способом. Множина  $F^e$  буде називатися базовою, а множина F – допоміжною.

Для мови  $\mathfrak{S}^e$  пропонується така семантика можливих світів:  $\mathcal{M} = \langle W, U, s, \varphi \rangle$  – модель мови  $\mathfrak{S}^e$  тоді й тільки тоді, коли: W – непорожня базова множина можливих світів; U – непорожня допоміжна множина можливих світів; s – функція типу:  $s: W \rightarrow U$ ;  $\varphi$  – функція приписування  $\varphi(p) = \langle TW(p), TU(p), FU(p) \rangle \forall p \in \text{Var}$ , де  $TW(p) \subseteq W$  і  $TU(p) \subseteq U$  – множина світів, де р істинно, а  $FU(p) \subseteq U$  – множина світів, де р хибно, причому  $TU(p) \cap FU(p) = \emptyset$ ,

але в системі світів  $U$  припускаються невідомі значення істинності  $p$ .

Інтенціональна система першого порядку має припущення, наміри, бажання тощо, які стосуються тільки елементів допоміжної множини  $F$ , тоді як інтенціональні системи вищих порядків мають припущення, наміри, бажання тощо стосовно елементів базової множини  $F^c$ . Через недостатню для моделювання поведінки ІА виразну потужність інтенціональних систем першого порядку потрібно використовувати інтенціональні системи вищих порядків. Для цього необхідно ввести правила декомпозиції модальних операторів за допомогою схем аксіом і правила виведення «modus ponens»: якщо  $\gamma, \gamma \rightarrow \delta$ , то  $\delta$ . Нехай  $\alpha$  і  $\beta$  належать допоміжній множині формул  $F$ ,  $\gamma$  і  $\delta$  – формули базової множини  $F^c$ ,  $i = \overline{1, N}$ .

**Аксіоми, які здійснюють опис модального оператора переконання.** Для  $N$  суб'єктів,  $i = \overline{1, N}$ ,  $Вір$  означає, що « $i$ -й суб'єкт вважає  $p$ ».

- A1.  $Ві(\alpha \ \& \ \beta) \equiv (Ві\alpha \ \& \ Ві\beta)$ ;
- A2.  $Ві(\alpha \ \vee \ \beta) \equiv (Ві\alpha \ \vee \ Ві\beta)$ ;
- A3.  $Ві(\alpha \ \rightarrow \ \beta) \equiv (Ві\alpha \ \rightarrow \ Ві\beta)$ ;
- A4.  $Ві \neg(\alpha \ \& \ \beta) \equiv (Ві \neg\alpha \ \vee \ Ві \neg\beta)$ ;
- A5.  $Ві \neg(\alpha \ \vee \ \beta) \equiv (Ві \neg\alpha \ \& \ Ві \neg\beta)$ ;
- A6.  $Ві \neg(\alpha \rightarrow \beta) \equiv (Ві\alpha \ \& \ Ві \neg\beta)$ ;
- A7.  $Ві \alpha \rightarrow \neg Ві \neg\alpha$ ;
- A8.  $Ві \alpha \rightarrow Ві \neg \neg\alpha$ ;
- A9.  $Ві \neg \neg\alpha \rightarrow Ві\alpha$ .

**Аксіоми, які здійснюють опис модального оператора знання.** Введемо в мову  $\mathcal{L}_e$  оператор знання  $K$  та оператор помилки  $E$ , що характеризують пізнання:  $Ki\alpha$  означає « $i$ -той суб'єкт знає  $\alpha$ », а  $Ei\alpha$  – « $i$ -той суб'єкт помиляється, припускаючи  $\alpha$ ».

A10.  $Ki\alpha \rightarrow (Ві\alpha \ \& \ \alpha)$ . У цій схемі використаний знак імплікації: при визначенні помилки достатньо зазначити хибність того, що суб'єкт приймає як істинне, але з того факту, що щось істинно для суб'єкта та істинно у фіксованій сис-

темі, ще не можна дійти висновку про те, що суб'єкт знає це.

A11.  $Ei\alpha \equiv (Ві\alpha \ \& \ \neg\alpha)$ . Має місце транзитивне відношення досяжності.

A12.  $Ki\alpha \rightarrow Ki \ Ki\alpha$ . Аксіома позитивної інтроспекції свідчить про усвідомлення агентом своїх знань.

A13.  $Ві\alpha \rightarrow Ki \ Ві\alpha$ .

**Аксіоми, які здійснюють опис модального оператора цілі.** Суб'єкт  $\alpha$  може мати несумісні бажання, і він може не вважати їх досяжними. Доцільніше використовувати поняття цілі – сумісної підмножини бажань:

- B1.  $Di(\alpha \ \& \ \beta) \equiv (Di\alpha \ \& \ Di\beta)$ ;
- B2.  $Di(\alpha \ \vee \ \beta) \equiv (Di\alpha \ \vee \ Di\beta)$ ;
- Цілі не суперечать одна одній:
- B3.  $Di \neg\alpha \equiv \neg Di \ \alpha$ ;
- B4.  $Di \neg(\alpha \ \& \ \beta) \equiv (Di \neg\alpha \ \vee \ Di \neg\beta)$ ;
- B5.  $Di \neg(\alpha \ \vee \ \beta) \equiv (Di \neg\alpha \ \& \ Di \neg\beta)$ ;
- B6.  $Di \ \alpha \equiv Di \neg \neg\alpha$ ;
- B7.  $Di (\alpha \rightarrow \beta) \rightarrow (Di \ \alpha \rightarrow Di \ \beta)$ ;
- B8.  $Ві(\alpha \rightarrow \beta) \ \& \ Di(\alpha) \rightarrow Di \ \beta$ ;
- B9.  $Ві(Di \ \alpha) \equiv Di(\alpha)$ .

**Аксіоми, які здійснюють опис модального оператора наміру:**

- C1.  $Ii(\alpha \ \& \ \beta) \equiv (Ii\alpha \ \& \ Ii\beta)$ ;
- C2.  $Ii(\alpha \ \vee \ \beta) \equiv (Ii\alpha \ \vee \ Ii\beta)$ .
- Наміри не суперечать один одному:
- C3.  $\neg Ii(\alpha) \equiv Ii \neg\alpha$ .
- C4.  $Ii \neg(\alpha \ \& \ \beta) \equiv (Ii \neg\alpha \ \vee \ Ii \neg\beta)$ ;
- C5.  $Ii \neg(\alpha \ \vee \ \beta) \equiv (Ii \neg\alpha \ \& \ Ii \neg\beta)$ ;
- C6.  $Ii \ \alpha \equiv Ii \neg \neg\alpha$ ;
- C7.  $Ii (\alpha \rightarrow \beta) \rightarrow (Ii \ \alpha \rightarrow Ii \ \beta)$ ;
- C8.  $Ii \ \alpha \rightarrow Di\alpha$ ;
- C9.  $Ві(Ii \ \alpha) \equiv Ii(\alpha)$ .

Вищезапропонований набір схем аксіом є несуперечливим, тобто для  $\forall \alpha \in F^c$  не виведена  $\alpha \ \& \ \neg\alpha$  (тому, що переконання, цілі й наміри кожного конкретного суб'єкта не суперечать один одному за визначенням). Водночас запропонована система аксіом загалом не повна,



тобто  $\forall \alpha \in F^e$ , для яких не виведені ні  $\alpha$ , ні  $\neg \alpha$  (це зумовлено неповнотою знань і переконань конкретного суб'єкта, який може не мати про якийсь конкретний факт жодної інформації).

За допомогою цих ІнВ відображаються компоненти моделі ІА: модель ПрО складається з переконань і знань ПА щодо концептів ПрО та зв'язків між ними; модель користувача містить переконання ПА, пов'язані з взаємодією, з користувачем. Моделі інших агентів дають змогу ПА використовувати для досягнення своїх намірів і цілей їх знання та переконання тощо.

Якщо ПА використовує онтологічну модель ПрО, тоді його переконання Ві використовують концепти з  $X = X_{cl} \cup X_{ind}$  та відношення  $R = r_{ier\_cl} \cup \{r_i\} \cup \{p_j\}$  формальної моделі онтології (1), що є специфічними для обраної ПрО, та мають відповідати всім обмеженням, що містяться у відповідній онтології.

Ці аксіоми можуть використовуватися для моделювання поведінки ПА, що входять до складу різних МПС. Розглянемо це на прикладі системи семантичного пошуку МАПС, архітектура якої описана на основі агентної парадигми.

### Застосування інтенціональних відношень в онтологічній моделі МАПС

МАПС – це метапошукова система з розвиненими засобами інтелектуалізації, що призначена для пошуку інформації у відносно вузьких предметних областях, пов'язаних з професійними чи науковими інтересами користувачів [23]. Основа МАПС – технологія Semantic Web, зокрема мова подання онтологій OWL, і засоби його обробки. Для подання знань, що цікавлять користувача, використовуються онтології ПрО та тезауруси задач, що базуються на них. При цьому тезаурус будується користувачем за відповідною онтологією самостійно, а онтологія вибирається з набору онтологій, запропонованих на сайті розробниками системи.

Система МАПС орієнтована на користувачів, що мають у мережі постійні

інформаційні інтереси та потребують постійного надходження відповідної інформації. Для цього МАПС надає можливість зберігати й повторно виконувати запити, з огляду на реакцію користувача на раніше запропоновані йому ІР (персональна фільтрація), відстежувати появу аналогічних запитів в інших користувачів (колаборативна фільтрація), зберігати формальний опис сфери інтересів користувача у вигляді онтології (семантична фільтрація) тощо.

Архітектура МАПС містить ПА трьох типів: ППА, що підтримують інтелектуальну та персоніфіковану взаємодію користувачів з системою; агентів ІР та агентів-посередників, що обслуговують взаємодію інших агентів [24].

В МАПС, як і в багатьох інших ПС, використовуються внутрішні та зовнішні онтології. До внутрішніх відноситься *онтологія семантичного пошуку*, які описує основні об'єкти МАПС (користувачів, ресурси, запити тощо), їх властивості та відношення між ними [25]. Ця внутрішня онтологія потрібна для побудови профілю користувача та містить такі основні класи:

- *користувач* – клас, що дозволяє описати властивості людини, що звертається до МАПС;

- *тема* – множина запитів, пов'язаних з однією інформаційною потребою

$$\text{thema} = \langle \text{id}_{\text{thema}}, \{z_q\}, q = \overline{1, u} \rangle,$$

що дає змогу поєднувати семантично пов'язані запити різних користувачів, які базуються на різних онтологіях і тезаурусах;

- *запит* – множина ключових слів, що характеризують інформаційну потребу користувача;

- *інформаційний ресурс* – відомості про ІР, раніше запропоновані користувачу, та їхні оцінки

$$\langle U_{url}, \{z_i, m_i, q_i\}, i = \overline{1, n} \rangle;$$

- *ІО* – відомості про інформаційний об'єкт з певною структурою, визначеною користувачем, що містяться в одному

чи декількох IP (структура IO може бути описана через посилання на відповідну зовнішню онтологію);

– *інформаційне середовище* – сукупність усіх доступних IP та IO, їх властивостей і зв'язків між ними;

– *тезаурус задачі* – множина термінів ПрО, що стосуються поточної задачі користувача

$$Th_{i,j} = \left\{ \left\langle th_{k_{ij}} \in T_{PrO_i}, v_{k_{ij}} \right\rangle, k = \overline{1, s_{ij}} \right\}, j = \overline{1, m_i};$$

– *результат запиту* – множина пар, першим елементом яких є посилання на IP, а другим – оцінки цих IP користувачем

$$rez = f(z, u) = \left\{ \langle id_{ir}, rating_{ir} \rangle \right\};$$

– *онтологія ПрО*, що містить опис області, до якої належать інформаційні потреби користувача

$$O_{PrO_i} = \langle T_{PrO_i}, R_{PrO_i}, F_{PrO_i} \rangle, i = \overline{1, n}.$$

Останній клас дозволяє пов'язувати різноманітні зовнішні онтології, які містять знання тих ПрО, що цікавлять користувачів, із цими користувачами та їх запитами: слід відмітити, що той самий користувач може застосовувати кілька онтологій для різних запитів та будувати за однією онтологією різні тезауруси відповідно до специфіки поточного завдання. Зовнішні онтології не залежать від розробників МАПС – вони будуються самими користувачами або експортуються із зовнішніх репозиторіїв онтологій. Тому неможливо заздалегідь накладати будь-які обмеження на ці онтології, а це ускладнює їх обробку та звужує можливості їх застосування. Тому важливо певним чином класифікувати такі зовнішні онтології (приміром, за виразною потужністю, методами побудови або джерелами).

Ще одна внутрішня онтологія, що застосовується в МАПС, – це *лексична онтологія ПрО*, що містить відомості про лексеми природних мов, які відповідають термінам онтології ПрО та використовується для встановлення зв'язків між елементами природномовних документів і термінами онтології ПрО

$$O_{lex_i} = \langle T_{lex_i}, R_{lex_i}, f \rangle, i = \overline{1, n},$$

тобто  $\forall t \in T_{PrO_i}$ .

Така онтологія може бути представлена мовою OWL і вживатися в різних застосуваннях, пов'язаних з використанням семантичного пошуку [25].

Розглянемо детальніше ППА, що використовуються в МАПС. Основні їх функції – це переформулювання запиту користувача з використанням відповідних онтологій та тезаурусів, передання цього запиту іншим ПА та перевпорядкування отриманої інформації відповідно до інформаційних потреб користувача.

ППА презентує інтереси користувача у взаємодії з системою, що відносяться до ПА. Для опису поведінки такого агента використовуються такі інтенціональні відношення, як знання, переконання і цілі користувача. Застосування такого формалізму, як ППА, дасть змогу, з одного боку, уникнути приписування людині-користувачу штучно обмеженої і формально схарактеризованої сфери інтересів, а з іншого – забезпечить засоби та методи прогнозування його вчинків у межах моделі взаємодії користувача та IP у відкритому інформаційному середовищі.

Для створення моделі, відповідно до методології розробки BDI-агентів, треба виконати наступні дії:

- визначити функції класів ПА, що входять до складу системи, та їх взаємодію;
- ідентифікувати тривалість життя ПА кожного класу та умови, за якими вони починають та припиняють свою діяльність;
- ідентифікувати інформаційний зміст взаємодії класів агентів; події й умови, що мають бути помічені; дії, що будуть виконуватися; інші інформаційні вимоги;
- побудувати класи агентів через успадкування або укрупнення, керуючись спільністю тривалості життя, інформації, інтерфейсів і послуг;
- описати для кожного класу ПА моделі їх переконань, цілей та намірів, навести конкретні класи та екземпляри агентів.

Ті переконання ППА  $V_{ППА_i}$ , що стосуються зовнішнього світу

$$V_{ППА_{ПрО_k}}, k = \overline{1, n}, \{V_{ППА_{ПрО_k}}\} \subseteq \{V_{ППА_i}\},$$

базуються на зовнішніх онтологіях МАПС. Ця множина не може бути порожньою, тому що тоді використання МАПС не матиме сенсу. Ці переконання можуть змінюватися в процесі взаємодії користувача з МАПС – приміром, користувач може обрати іншу онтологію ПрО та будувати за нею нові тезауруси та запити. Тоді оцінка IP, обраних раніше з урахуванням інших переконань, може змінюватися. Переконання ППА містять також елементи, побудовані МАПС на основі аналізу взаємодії з користувачем:

$$V_{ППА_{search_m}}, m = \overline{0, p}, \{V_{ППА_{search_m}}\} \subseteq \{V_{ППА_i}\}.$$

Ця множина на початку взаємодії може бути порожньою і поповнюється в процесі роботи. Користувач може явно або за певним шаблоном видаляти ті елементи свого досвіду, які не відповідають його поточним інтересам (приміром, після закінчення проекту або зміни місця роботи). Така інформація буде зберігатися як екземпляри відповідних класів онтології семантичного пошуку.

Знання ППА  $K_{ППА_j}$  базуються на внутрішніх онтологіях МАПС та не змінюються на протязі використання системи. У логічному виведенні можуть використовуватися як знання, так і переконання ППА. Знання можуть стосуватися структури IO, що потрібний користувачеві, а переконання – його семантики, значень властивостей та зв'язків з іншими екземплярами IO. При цьому не має значення, наскільки онтологія ПрО відповідає реальному світові, але важливо, щоб вона була пертинентна задачі користувача та не містила внутрішніх логічних суперечностей. Приміром, користувач може досліджувати певні соціологічні аспекти взаємин між різними расами у фентезі, і, хоча насправді ельфи та орки не існують, використовувати їх формалізовані описи, побудовані за літературними творами.

Цілі ППА  $D_{ППА_k}$  можна поділити на основні та допоміжні. Основні відповідають виконанню завдань ППА, а допоміжні дозволяють підвищити ефективність його функціонування. У формулюванні цілей використовуються терміни як онтології семантичного пошуку (“IO”, “запит” тощо) та їх властивості (“час виконання запиту”, “тип IO” тощо), так і онтології ПрО (приміром, щоб замінити термін запиту на його надклас чи підклас).

Допоміжні цілі пов'язані зі здобуттям неявних переваг та вимог користувача, що логічно впливають з оцінок результатів запитів (наприклад, узагальнити результати багаторазових запитів і видобути їх спільні характеристики, які користувач не вказує явно, щоб потім додавати їх до запиту автоматично); з групуванням користувачів із подібними інтересами (приміром, тих, що використовують схожі онтології або подібні запити). Виконання допоміжних цілей може викликати зміни у множині переконань користувача: приміром, спростити онтологію ПрО шляхом відкидання термінів, що ніколи не використовуються у пошуку або доповнити онтологію новими термінами та зв'язками між ними, здобутими з документів – результатів пошуку.

## Висновки

Слід відмітити, що, якщо інтеграція BDI-моделі з елементами внутрішніх онтологій виконується самими розробниками кожної МПС, то використання елементів зовнішніх онтологій пов'язане з певними труднощами: невідома ані структура такої онтології, ані її складність. Це може призвести до помилок у роботі МІІС. Приміром, в переконаннях ППА передбачається, що в разі неуспішного виконання пошуку потрібно замінити певні властивості IO на властивості його надкласу. Але, якщо використовується онтологія ПрО з циклами в ієрархії класів, то це викличе зациклення алгоритму. Тому досить важливо мати можливість оцінювати зовнішні онтології ПрО та мати можливість для визначення характеристик окремих випадків онтологій. Прикладами

таких окремих випадків є розглянуті вище Wiki-онтологія та таксономія. Доцільно надалі виділити більшу кількість таких окремих випадків та формально описати їх характеристики в рамках заданої формальної моделі.

1. Gruber T.R. Toward Principles for the Design of Ontologies Used for Knowledge Sharing. *International Journal of Human-Computer Studies*. 1995. Vol. 43, Issues 5-6. P. 907–928.
2. Obr L., Ceuster W., Mani I., Ra S., Smith B. The evaluation of ontologies. In *Semantic web: Revolutionizing Knowledge Discovery in the Life Sciences*, New York: Springer Verlag, 2006. P. 139–158. <https://philpapers.org/archive/OBRTEO-6.pdf>.
3. OWL Web Ontology Language Semantics and Abstract Syntax. Section 2. Abstract Syntax [Електронний ресурс]. Режим доступу: <http://www.w3.org/TR/owl-semantics/syntax.html>.
4. Гладун А.Я., Рогушина Ю.В. Семантичні технології: принципи та практики. К.:ТОВ "ВД "АДЕФ-Україна". 2016. 308 с.
5. Protégé [Електронний ресурс]. Режим доступу : <http://protege.stanford.edu/>.
6. Рогушина Ю.В. Семантичний пошук у Web на основі онтологій: розробка моделей, засобів і методів Мелітополь: МДУПУ ім. Богдана Хмельницького, 2015.
7. Rogushina J. Semantic Wiki resources and their use for the construction of personalized ontologies. *CEUR Workshop Proceedings 1631*. 2016. P. 188–195.
8. Shoham Y. Agent-oriented programming. *Artificial Intelligence*. 1993. N 60. P. 51–92.
9. Franklin A., Graesser A. Is it an agent or just a programm: a taxonomy form autonomos agents. In *Proc. of the III Int. Workshop on Agents Theories, Arch. and Languages*. N.-Y., Springer-Verlad, 1996.
10. Рогушина Ю.В. Программные агенты: определения, таксономии, модели. *Управляющие системы и машины*. 2001. № 5. С. 39–45.
11. Griss M.L. Software agents as next generation software components. *Component-based software engineering*. 2001. P. 641–657. <https://pdfs.semanticscholar.org/839e/b7a89e4502b637fc6ad5a46acd4dca0cad40.pdf>.
12. Petrie C.J. Agent-based ingeneering, the WEB and intelligent.IEEE expert. 1996. 11(6). P. 24–29.
13. Busuioc M., Winter C. Negotiation and Intelligent Agents, Project NOMADS-001, BT Labs, Martlesham Heath, U.K., 1995.
14. Giménez-Lugo G.A., Amandi A., Sichman J.S., Godoy D. Enriching information agents' knowledge by ontology comparison: A case study. In *Ibero-American Conference on Artificial Intelligence, 2002*, Springer Berlin Heidelberg. P. 546–555.
15. Lieberman H., Selker T. Agents for the user interface. *Handbook of Agent Technology*. 2003. P. 1–21.
16. Lahlouhi A., Sahnoun Z., Lamine Benbrahim M., Boussahal A. Interface agents development in MASA for human integration in multiagent systems. *Advances in Artificial Intelligence, IBERAMIA 2002*. P. 566–574.
17. Konolige K. A Deduction Model of Belief . *Proc. of the 1986 Workshop on Reasoning About Actions and Plans*. 1987. P. 297–340.
18. Newell A. The knoledge level. *Artificial Intelligence*. 1982. N 18. P. 87–127.
19. Cohen P.R., Levesque H.J. Intention is choice with commirment. *Artificial Intelligence*. N 42. 1990. P. 213–261.
20. Rao A.S., Georgeff M.P. Modeling rational agents within a BDI-architecture. In R. Pikes and E. Sandewall, eds. *Proc. of Knowledge Representation and Reasoning (KR&R-91)*, Morgan Kaufmann Publishers: San Mateo, CA. 1991. P. 473–484.
21. Bratman M.E., Israel D., Pollack M.E. Toward an architecture for resource-bounded agents. *Technical Report CSLI-87-104*, Center for the Studv of Language and Information, SRI and Stanford University, 1987.
22. Плескач В.Л., Рогушина Ю.В. Агентні технології. Монографія, К.: Київ. нац. торг.–екон. ун-т, 2005. 338 с.
23. Rogushina J.V. The Use of Ontological Knowledge for Semantic Search of Complex Information Objects. *OSTIS-2017, Minsk, BGUIR*. 2017. P. 127–132.
24. Rogushina J. Use of the Ontological Model for Personification of the Semantic Search. *International Journal of Mathematical Sciences and Computing(IJMISC)*, Vol. 2, N 1. 2016. P. 1–15. <http://www.mecs-press.org/ijmsc/ijmsc-v2-n1/IJMISC-V2-N1-1.pdf>.
25. Рогушина Ю.В., Гришанова І.Ю. Літературний твір наукового характеру "Модель мультиагентної інформаційно-пошукової системи МАПС (Модель МАПС)". Сві-

доцтво про реєстрацію авторського права на твір № 32068.

26. Rogushina J. Methods and tools of knowledge management at the Semantic Web environment. *International Journal «Information Theories and Applications»*. Vol. 19, N. 3. 2012. P. 258–268. <http://www.foibg.com/ijita/vol19/ijita19-3-p08.pdf>.

## References

1. Gruber T.R. (1995) Toward Principles for the Design of Ontologies Used for Knowledge Sharing. *International Journal of Human-Computer Studies*. Vol. 43, Issues 5-6. P. 907–928.
2. Ohrs L., Ceuster W., Mani I., Ra S., & Smith B. (2006) The evaluation of ontologies. In *Semantic web: Revolutionizing Knowledge Discovery in the Life Sciences*, New York: Springer Verlag, 139–158. <https://philpapers.org/archive/OBRTEO-6.pdf>.
3. OWL Web Ontology Language Semantics and Abstract Syntax. Section 2. Abstract Syntax [Online]. <http://www.w3.org/TR/owl-semantics/syntax.html>.
4. Gladun A.Ya., Rogushina J.V. (2016) *Semantic technologies: principles and practices*. Kyiv, ADEF-Ukraine, 308 p. (in Ukrainian).
5. Protégé [Online]. <http://protege.stanford.edu/>.
6. Rogushina J.V. (2015) Semantic retrieval in the Web on base of ontologies: design of methods, means and methods. Melitopol, MDPU. (in Ukrainian).
7. Rogushina J. (2016) Semantic Wiki resources and their use for the construction of personalized ontologies. *CEUR Workshop Proceedings 1631*. P. 188–195.
8. Shoham Y. Agent-oriented programming. *Artificial Intelligence*. 1993. N 60. P. 51–92.
9. Franklin A., Graesser A. (1996) Is it an agent or just a program: a taxonomy form autonomous agents. In *Proc. of the III Int. Workshop on Agents Theories, Arch. and Languages*. N.-Y., Springer-Verlag.
10. Rogushina J. (2001) Software agents: definitions, taxonomies, models. *Control systems and machines*. N 5, P. 39–45. (in Russian).
11. Griss M.L. (2001) Software agents as next generation software components. *Component-based software engineering*. P. 641–657. <https://pdfs.semanticscholar.org/839e/b7a89ed502b637fc6ad5a46acd4dca0cad40.pdf>.
12. Petrie C.J. (1996) Agent-based ingeneering, the WEB and intelligent.IEEE expert, 11(6), P. 24–29.
13. Busuioc M., & Winter C. (1995) *Negotiation and Intelligent Agents*, Project NOMADS-001, BT Labs, Martlesham Heath, U.K.
14. Giménez-Lugo G.A., Amandi A., Sichman J.S., & Godoy D. (2002) Enriching information agents' knowledge by ontology comparison: A case study. In *Ibero-American Conference on Artificial Intelligence*, Springer Berlin Heidelberg. P. 546–555.
15. Lieberman H., & Selker T. (2003) Agents for the user interface. *Handbook of Agent Technology*. P. 1–21.
16. Lahlouhi A., Sahnoun Z., Lamine Benbrahim M., & Boussaha1 A. (2002) Interface agents development in MASA for human integration in multiagent systems. *Advances in Artificial Intelligence, IBERAMIA*. P. 566–574.
17. Konolige K. (1987) A Deduction Model of Belief. *Proc. of the 1986 Workshop on Reasoning About Actions and Plans*, P. 297–340.
18. Newell A. The knoledge level. *Artificial Intelligence*. 1982. N 18. P. 87–127.
19. Cohen P.R., & Levesque H.J. (1990) Intention is choice with commirment. *Artificial Intelligence*. N 42. P. 213–261.
20. Rao A.S., & Georgeff M.P. (1991) Modeling rational agents within a BDI-architecture. In R. Pikes and E. Sandewall, eds. *Proc. of Knowledge Representation and Reasoning (KR&R-91)*, Morgan Kaufmann Publishers: San Mateo, CA. P. 473–484.
21. Bratman M.E., Israel D., & Pollack M. E. (1987) Toward an architecture for resource-bounded agents. *Technical Report CSLI-87-104*, Center for the Studv of Language and Information, SRI and Stanford University .
22. Pleskach V.L., & Rogushina J.V. (2005) *Agent technologies*. Kyiv. KNTU. 338 p. (in Ukrainian). <https://core.ac.uk/download/pdf/38468943.pdf>
23. Rogushina J.V. (2017) The Use of Ontological Knowledge for Semantic Search of Complex Information Objects. *OSTIS-2017*. Minsk. BGUIR. P. 127–132.
24. Rogushina J. (2016) Use of the Ontological Model for Personification of the Semantic Search. *International Journal of Mathematical Sciences and Computing (IJMSC)*, Vol. 2, N 1. P. 1–15. <http://www.mecs-press.org/ijmsc/ijmsc-v2-n1/IJMSC-V2-N1-1.pdf>.

25. Rogushina J., & Grishanova I. The literary work of a scientific nature "Model of multiagent information retrieval system MAIPS" (MAIPS Model)". Certificate of registration of copyright № 32068.
26. Rogushina J. (2012) Methods and tools of knowledge management at the Semantic Web environment. International Journal «Information Theories and Applications», Vol. 19, N 3. P. 258–268. <http://www.foibg.com/ijita/vol19/ijita19-3-p08.pdf>.

Одержано 18.04.2017

***Про автора:***

*Рогущина Юлія Віталіївна,*  
кандидат фізико-математичних наук,  
старший науковий співробітник.  
Кількість наукових публікацій в  
українських виданнях – 140.  
Кількість наукових публікацій в  
зарубіжних виданнях – 30.  
Індекс Хірша – 10.  
<http://orcid.org/0000-0001-7958-2557>.

***Місце роботи автора:***

Інститут програмних систем  
НАН України,  
03181, Київ-187,  
проспект Академіка Глушкова, 40.  
Тел.: 066 550 1999.  
E-mail: [ladamandraka2010@gmail.com](mailto:ladamandraka2010@gmail.com)