

*Integration of Information From Semi-Structured Web Data Sources on the Basis of Ontology and Meta-Models, 7th International Baltic Conference on Databases and Information Systems Vilnius, Lithuania, 2006, P. 177 – 186.* 5. Levy A.Y. *Logic-Based Techniques in Data Integration. Logic-based Techniques in Data Integration. In: Logic Based Artificial Intelligence. Edited by J. Minker. Kluwer Publishers, 2000.* 6. Manolescu I., Florescu D., Kossman D. *Answering XML Queries over Heterogeneous Data Sources. Proc. Of the 27th VLDB Conference, Roma, Italy, 2001.*

УДК 004.89

В.В. Литвин

Національний університет “Львівська політехніка”,  
кафедра інформаційних систем та мереж

## МЕТОД ВИКОРИСТАННЯ ОНТОЛОГІЙ У ПЕТЛІ OODA

© Литвин В.В., 2014

Досліджено поведінку інтелектуального агента у конкурентному середовищі. Для моделювання поведінки обрано петлю OODA. Розглянуто взаємодію етапів петлі OODA (спостереження, орієнтація, прийняття рішення, дія) із онтологією задач та предметної області, в межах якої функціонує цей агент.

**Ключові слова:** петля OODA, спостереження, орієнтація; аналіз; синтез; прийняття рішення; дія, онтологія.

In the paper the behavior of an intelligent agent in a competitive environment is investigated in the paper. The OODA loop is chosen for behavior simulation. The interaction of OODA loop stages (observation, orientation, decision support, action) with the ontology of tasks and subject area in which the agent operates was explored.

**Key words:** OODA loop, observation, orientation, analysis, synthesis, decision support, action, ontology.

### Вступ. Постановка проблеми у загальному вигляді

Відповідно до ідей Джона Бойда та його послідовників будь-яку діяльність у конкурентному середовищі (наприклад, у військовій сфері) з певним ступенем наближення можна подати у вигляді кібернетичної моделі OODA [1]. Зазначена модель передбачає багаторазове повторення петлі дій, складеної з чотирьох послідовних взаємодіючих процесів (рис. 1.): спостереження (observation), орієнтація (orientation), прийняття рішення (decision), дія (action).

Така модель з успіхом почала застосовуватися для моделювання діяльності та прийняття рішень у бізнесі, політиці, соціології тощо, тобто у тих сферах, де наявна конкуруюча сторона.

Згідно з теорією Бойда, кожна людина або організація при вирішенні поставлених перед ними завдань має свою петлю прийняття рішень і діяльності.

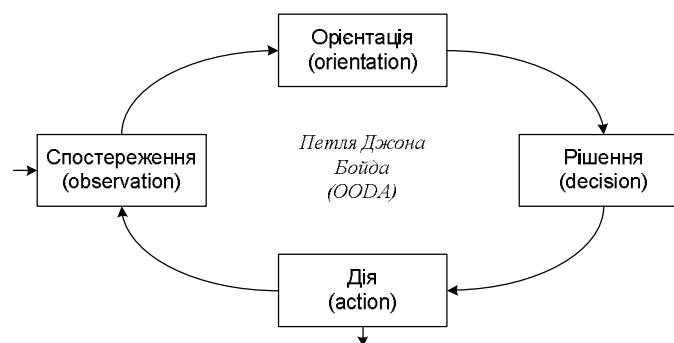


Рис. 1. Процеси петлі OODA

## Аналіз останніх досліджень та публікацій

Згідно з літературними джерелами з теорії Д. Бойда, підсумок з яких наведено в [1], мета етапів петлі та їх функціональне призначення є такими.

*Спостереження* (observation) – це процес збирання інформації, необхідної для прийняття рішення у деякому конкретному випадку. Необхідна інформація може бути отримана як від зовнішніх, так і від внутрішніх джерел. Під внутрішніми джерелами інформації розуміються елементи зворотного зв'язку петлі. Як зовнішні використовуються датчики, а також інші канали отримання інформації.

Щоб спостереження набуло наукового характеру, потрібно, щоб воно:

- було планомірним, а не випадковим;
- здійснювалося послідовно й систематично;
- було забезпечене достатньо широкою інформацією про явище, яке є предметом спостереження (слід оперувати якомога більшою кількістю фактів);
- передбачало точну фіксацію результатів спостереження.

Збирання даних може здійснюватися:

1) механічним способом; механічна реєстрація даних полягає в тому, що джерело інформації, тобто “подія” або “явище”, виявляється у вигляді зміни деякого фізичного стану, і цей новий стан реєструється механічним способом.

2) експертом; спостереження, здійснюване експертом, із подальшим відновленням результатів по пам'яті, які називають “записом”.

3) шляхом експериментального дослідження, особливість якого полягає в тому, що явище (предмет дослідження) вивчається за різних умов та обставин; застосування цього методу дослідження сприяє глибокому і дуже точному вивченню певної психологічної закономірності.

*Орієнтація* (orientation) – найвідповідальніший і найскладніший з когнітивного погляду етап у всьому циклі OODA. Етап орієнтації складається з двох підетапів: руйнування (destruction) і творення (creation). Руйнування передбачає розбиття ситуації на дрібні елементарні частини, які легші для розуміння. Людина або організація, яка приймає рішення, намагатиметься декомпонувати завдання до такого рівня, поки новоутворені складові завдання не стануть близькими до стандартних або типових ситуацій, для яких об'єкт прийняття рішень (ОПР) має план дій. Ознайомлюються з цими елементарними типовими підзадачами за допомогою навчання, тренування, накопичення досвіду та інструктажу. ОПР ідентифікує поточну ситуацію, тобто відносить її до певної підзадачі і застосовує заздалегідь заготовлений план дій для цієї підзадачі. Потім ці складові елементарні підплани об'єднуються в загальний план дій, що і відповідає підетапу “творення”. Якщо немає планів, серед яких можна обрати рішення, то процес залишається на етапі орієнтації, і здійснюється подальше декомпонування задачі. Якщо не вдається розробити план з реальними шансами на успіх, то подальше подрібнення може призвести до останнього циклу.

Для орієнтації використовуються методи аналізу і синтезу, які тісно пов'язані між собою. Вони призначені для обробки інформації, отриманої в результаті застосування дослідницьких методів.

Аналіз являє собою вивчення якостей, властивостей і характеристик досліджуваного об'єкта за допомогою його умовного поділу на окремі складові.

Своєю чергою, синтез полягає в узагальненні інформації про окремі складові і формуванні сукупності інформаційних даних про об'єкт дослідження загалом.

Результати, отримані у процесі аналізу та синтезу, є основою для складання різного роду прогнозів на найближчу і далеку перспективу. Прогнозувати можна методами розрахунку та екстраполяції.

На етапі декомпозиції системи здійснюється:

- визначення та декомпозиція загальної мети дослідження й головної функції системи як обмеження траєкторії у просторі станів системи або в області допустимих ситуацій. Найчастіше декомпозицію виконують побудовою дерева цілей та дерева функцій;

- виділення системи із середовища (поділ на “систему” та “несистему”);
- опис впливових факторів;
- опис тенденцій розвитку;
- опис системи як “чорні скриньки”;
- функціональна (за функціями), компонентна (за типом елементів), структурна (за типом відношень між елементами) декомпозиція системи.

*Прийняття рішення* (decision) – третій етап циклу OODA. Якщо до цього етапу ОПР змогла сформувавши тільки один реальний план, то приймається рішення – виконувати цей план чи ні. Якщо ж сформовано кілька альтернативних варіантів дій, то ОПР на цьому етапі вибирає найкращий з них для подальшої реалізації. Вибирати найкращий план можна за критерієм ефективність–вартість. В умовах ліміту часу кращим вважається той план, що відповідає вимогам швидкої надійності.

Для прийняття рішення використовують такі методи:

- метод ефективність-вартість враховує три етапи: побудова моделі ефективності, побудова моделі вартості, синтез вартості й ефективності;
- методи теорії і практики надійності ґрунтуються на застосуванні апарата теорії ймовірностей і випадкових процесів, математичної статистики та моделювання.

*Дія* (action) – завершальний етап циклу, що передбачає практичну реалізацію обраного курсу дій або плану.

Існують два основні способи досягнення конкурентних переваг при здійсненні різних видів професійної діяльності. Перший шлях – зробити в кількісному вимірі свої цикли дій швидшими. Це дає нам змогу першими приймати рішення і змусить конкурентів реагувати на наші дії. Другий шлях – покращити якість прийнятих рішень, тобто приймати рішення, які більшою мірою відповідають ситуації, яка склалася, ніж рішення конкурентів.

Підвищити якість власних рішень можна різними способами, зокрема застосуванням сучасних формальних математичних методів, автоматизованих систем керування, систем підтримки прийняття рішень, експертних систем. Якщо використовувати останні, то за сучасним підходом до їх побудови використовують як ядра баз знань онтології [2]. Тому виникає задача розроблення методів використання онтологій у петлі OODA.

### **Формування цілей**

Розробити підхід до підвищення ефективності діяльності об’єкта управління в конкурентному середовищі. Задачею дослідження є розроблення та обґрунтування методу моделювання петлі OODA з використанням онтологій баз знань середовища, в якому функціонує об’єкт управління.

### **Основний матеріал**

Відмітна риса циклу OODA від інших циклічних моделей полягає в тому, що в будь-якій ситуації завжди передбачається наявність конкурентної сторони. На рис. 2 наведено три об’єкти управління, які знаходяться в деяких своїх початкових станах і мають власні стани мети. Те, що ці об’єкти функціонують у конкурентному середовищі, змушує під час проходження петлі OODA аналізувати стани, в яких знаходяться конкуренти та їх дії, що й відображено на рисунку відповідними стрілками.

Пропонується для моделювання петлі OODA використовувати інтелектуальну систему, ядром бази знань якої є онтологія, що складається із онтології предметної області (ПО) й онтології задач, які можуть виникати у цьому середовищі. На нашу думку, зміст онтології напряму впливає на 2-й і 3-й етапи циклу, а сама структура та наповнення онтології залежить від 1-го та 2-го етапів (рис. 3).

Розглянемо детальніше кожний етап петлі OODA у процесі його взаємодії з онтологією предметної області та задач, які в цій області виникають.

Етап спостереження дає змогу здійснювати процес розбудови онтології, а також аналізувати її з метою вибору релевантної інформації, яка потрібна на наступних етапах петлі OODA.

Формально онтологія складається з термінів (понять, концептів), організованих у таксономію, їх визначень і атрибутів, а також пов'язаних з ними аксіом і правил виведення. Тому під моделлю онтології  $O$  розуміють трійку вигляду [2]

$$O = \langle C, R, F \rangle,$$

де  $C$  – скінченна множина понять (концептів, термінів) ПО;  $R$  – скінченна множина відношень між концептами (поняттями, термінами) заданої ПО;  $F$  – інтерпретація понять та відношень (аксіоми). Аксіоми встановлюють семантичні обмеження для системи понять та відношень [3].

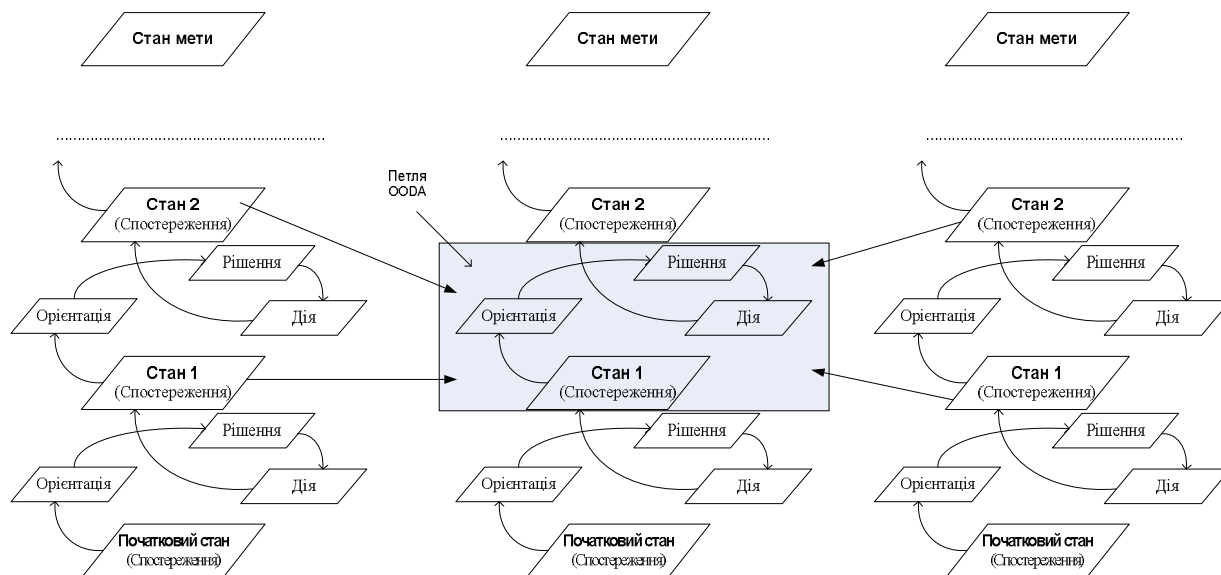


Рис. 2. Функціонування об'єктів у конкурентному середовищі

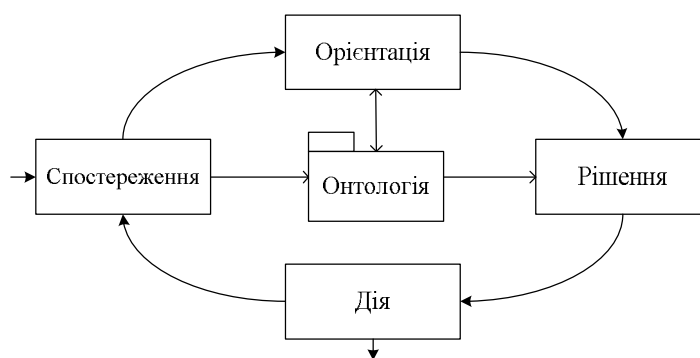


Рис. 3. Використання онтологій у петлі OODA

Для побудови онтології, яка адекватно описує семантичну модель ПО, необхідно насамперед розв'язати задачі одержання знань із різних джерел для виявлення множини концептів і встановлення ієрархії на цій множині. Оскільки значна частина інформації міститься в текстах природними мовами (ПМТ), перспективним є одержання знань із текстової інформації, а також інтелектуальне опрацювання спеціально підібраних колекцій ПМТ.

Одним із найефективніших підходів до наповнення онтології є її автоматизоване навчання природномовними текстами. Автоматизоване наповнення можна реалізувати за допомогою аналізу текстових документів, застосувавши процесор знань (рис. 4). Детальніше такий підхід розглянуто у монографії [4].

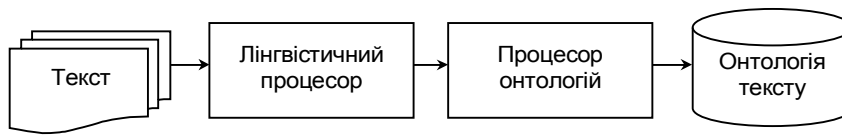


Рис. 4. Структурно-функціональна схема процесора знань

У поданій схемі завдання лінгвістичного процесора: виконати його лексичний, лексико-граматичний, синтаксичний та семантичний аналіз. У результаті цього онтологія поповнюється поняттями, СДО-трійками (суб'єкт – дія – об'єкт) і причинно-наслідковими зв'язками між СДО-трійками. Іншу частину важливих зв'язків між поняттями та їхніми властивостями встановлює процесор онтологій, котрий будує онтологічну структуру для кожного концепту  $C_i$ , отриманого після аналізу тексту. Робота процесора онтологій підтримується відповідною БЗ, основними компонентами якої є: по-перше, множина правил, по-друге, універсальна логічна БД MModWN [5] типу WordNet [6]. Процесор знань застосовують у системі автоматизованого одержання знань із текстових документів, яка, своєю чергою, застосовується для розв'язання задачі семантичного пошуку в повнотекстових БД. Серед систем, розроблених в Україні, треба зазначити розробку колективу кафедри математичної інформатики Київського національного університету імені Тараса Шевченка – систему опрацювання текстів природною мовою [7]. Система створена для розв'язування таких задач, як аналіз та синтез текстів природною мовою, автоматизоване генерування реферату тексту, автоматизована індексація (визначення тематики) тексту. Розроблена нами система детально описана в [4, 8]. Основна перевага нашого підходу полягає у побудові інтелектуального агента (ІА), який визначає цінність повідомлень, що пропонується додавати в онтологію залежно від вибраного плану управління.

Особливості функціонування спеціалізованого ІА визначаються його інтересом – вектором оцінок бажаності можливих станів агента. Для опису інтересу агента, за допомогою якого він розрізняє стани довколишнього світу та позиціонує себе у ньому, застосовується функція корисності, яка є числовою оцінкою його бажаності для агента. Корисності об'єднуються з імовірностями дій для визначення очікуваної корисності кожної дії.

Нехай  $U(S)$  – корисність стану  $S$  з погляду агента, що приймає рішення щодо вчинення деякої дії  $A$ . Довільна недетермінована дія може спричинити результуючий стан  $Result_i(A)$ , де індекс  $i$  пробігає по усіх можливих результатах. Перш ніж вчинити дію  $A$ , агент оцінює імовірність  $P(Result_i(A)|Do(A),E)$  кожного з можливих результатів, де  $E$  – сукупність доступних агенту параметрів його стану, а  $Do(A)$  – висловлювання, згідно з яким в поточному стані виконується дія  $A$ . Отже, можна обчислити умовну корисність дії  $EU(A|E)$  з урахуванням відомих параметрів стану:

$$EU(A|E) = \sum P(Result_i(A)|Do(A),E) \cdot U(Result_i(A)).$$

Якщо раціональний інтелектуальний агент керується принципом максимальної очікуваної корисності (Maximum Expected Utility – *MEU*), він змушений вибирати дію, яка максимізує очікувану корисність для агента. Так функціонує механізм мотивації поведінки раціонального інтелектуального агента незалежно від сфери його застосування.

У випадку інформаційно-пошукового агента його інтерес може бути заданий через оцінку новизни отриманих повідомлень, яка потребує застосування методів інтелектуального аналізу природномовних текстів. Вважаємо, що текст побудований як повідомлення. Структура повідомлення орієнтована на сприйняття іншим агентом, тому складається з двох частин (рис. 5.): констатуючої частини, за якою адресат оцінює релевантність повідомлення (1) та визначає його контекст (2), та конструктивної частини – потенційно нових для читача знань у даному контексті (3).



Рис. 5. Розпізнавання нових знань у повідомленні з метою наповнення ними бази знань

Якщо нове знання є не повним алгоритмом, а лише окремим фактом чи правилом, що вносить уточнення до вже відомих агенту алгоритмів, зміна їх функції корисності є оцінкою новизни цього факту (правила) та їх важливості для агента. Детальніше такий підхід описано у монографії [9]. Представлення знань у формі онтології передбачає, що будь-яке можливе узагальнення, тобто комплексне, складене поняття завжди явним чином артикульоване, назване і як окремий концепт фігурує в базі знань.

На етапі орієнтації будується план дій. Для досягнення цільового стану ІА насамперед повинен бути побудований план досягнення цього стану із всіма можливими альтернативами [10]. Планування ґрунтується на декомпозиції. Задача планування  $ZP$  містить три складові: множину станів  $S$ , множину дій  $A$ , множину цільових станів  $Goal$  (станів мети), тобто

$$ZP = \langle St, A, Goal \rangle.$$

Надалі вважатимемо, що стан мети єдиний. Якщо станів мети декілька, то мету можна записати як диз'юнкцію цих станів. Тоді досягнення такого стану є розв'язком деякої підзадачі, тому припущення про єдиність стану мети є нормальним.

Своєю чергою, дія  $A$  складається із чотирьох частин: ім'я дії, список параметрів, передумова та результат. А сам план визначається як кортеж із чотирьох елементів – <Множина дій, Множина обмежень впорядкування, Множина причинних зв'язків, Множина відкритих передумов> [11].

Стан  $S(i)$  задається у вигляді множини фактів із відповідними ймовірнісними оцінками. Дія  $a_{ij}$  подається у вигляді відображення зі стану  $S(i)$  в стан  $S(j)$  з відповідною ймовірністю  $p_{ij}$ , тобто  $S(i) \rightarrow S(j)$  з імовірністю  $p_{ij}$ .

Для вибору необхідних дій ІА повинен вміти оцінювати стани. Легше це здійснити зі станами, в яких він вже перебував. Важче оцінити майбутні стани. Для оцінювання використовуються евристичні функції або метазнання. Тому спочатку розглянемо оцінку пройдених станів, потім дій і насамкінець їхню комбінацію, що веде до нового (майбутнього) стану.

Нехай  $v(S(i))$  – оцінка стану  $S(i)$ . Для оцінювання станів, у яких вже перебував ІА, використовуватимемо онтологію ПО. Стан мети  $Goal$  визначається необхідністю деякій множині ознак  $X$  досягнути певних значень  $z(x, Goal) \forall x \in X$ . Будь-який стан  $S(i)$  задається своєю множиною ознак  $Y_i$ , які набувають значень  $z(y, S(i)) \forall y \in Y_i$ .

Для оцінювання стану  $S(i)$  необхідно здійснити відображення у множини ознак та їх значень стану  $S(i)$  у множину ознак та значень стану  $Goal$ . Очевидно, що таке відображення повинне використати БЗ, а саме додатковий модуль онтологій Semantic Web Rule Language (SWRL).

$$y : Y_i \xrightarrow{\hat{o}} X.$$

Тоді оцінка стану  $v(S(i))$  обчислюється

$$v(S(i)) = d(S(i), Goal) = \sum_{x \in X_W} j(z(y(y), S(i)), z(x, Goal)),$$

де  $X_W$  – множина найсуттєвіших ознак. Наприклад, для їх визначення можна задати вагу елементів онтології [4, 12];  $j$  – деяка метрика, яка залежить від специфіки ПО [4, 12].

У наших дослідженнях для вибору дій ІА ми ґрунтуватимемо на раціональності агента як прагнення мінімізувати витрати ресурсів для досягнення кінцевого стану. Тому вважатимемо, що кожна дія  $a_{ij}$  однозначно визначається витратами ресурсів  $g_{ij}^k$  (ціна переходу зі стану в стан), де  $k=1,2,\dots,n_i$ .  $n_i$  – кількість альтернатив  $a_k$  для здійснення переходу  $a_{ij}$ . Тому надалі дію позначатимемо трьома індексами  $a_{ij}^k$ : перехід зі стану  $S(i)$  у стан  $S(j)$ , використовуючи альтернативу  $a_k$  [13].

Оскільки що оцінка менша, то краще, то оцінка дії прямо пропорційна витраті ресурсів, тобто:

$$v(a_{ij}^k) = E \cdot g_{ij}^k,$$

де  $E$  – скалярна величина, яка зводить вимірювання оцінки дії до одного вимірювання з оцінкою станів.

Найкращий план вибирають на етапі рішення. Загалом рішення стосовно вибору дії на основі альтернативи приймаємо згідно з деяким відношенням між станом та дією:

$$o_i(a_{ij}^k) = d(v(a_{ij}^k), v(S(j))).$$

Зокрема таке відношення може бути лінійним:

$$o_i(a_{ij}^k) = wv(a_{ij}^k) + (1-w)v(S(j)),$$

де  $w \in [0,1]$  – частка альтернативі дії, яку ІА віддає під час прийняття рішення, інша частка належить стану, в який він перейде.

Після оцінювання дій та станів задача вибору шляху зводиться до задачі асинхронного динамічного програмування [14]. Ми отримаємо таку модель переходів між станами:

$$S(j) = a(S(i), o_i) \quad (1)$$

із критерієм оптимізації

$$\Theta(S(0), \mathbf{o}) \Rightarrow \min(\max). \quad (2)$$

Задача (1)–(2) є задачею динамічного програмування. Використовуючи методи, придатні для розв’язування таких задач, знаходимо розв’язок у вигляді шляху переходу з початкового у кінцевий стан, тобто шлях виконання плану.

У подальших наших роботах планується ускладнити модель (1)–(2) з урахуванням станів конкурентів і використати розроблене математичне забезпечення для моделювання воєнних дій.

### Висновки

У роботі розглянуто моделювання поведінки інтелектуального агента, який функціонує у конкурентному середовищі. Для моделювання поведінки обрано петлю OODA. Для підвищення ефективності циклу OODA запропоновано використати онтологію предметної області, в межах якої функціонує інтелектуальний агент, та онтологію задач, які виникають у цій області. Визначено вплив етапів петлі OODA на зміст онтології й навпаки – зміст онтології на перебіг проходження етапів. Запропоновано взаємодію між онтологією та етапами петлі OODA. Задачу планування діяльності інтелектуального агента у конкурентному середовищі зведено до задачі динамічного програмування.

1. Ивлев А.А. Основы теории Бойда. Направления развития, применения и реализации / Ивлев А.А. – М., 2008. – 74с. 2. Gruber T. A translation approach to portable ontologies / T. Gruber // Knowledge Acquisition. – 1993. – N 5 (2). – P. 199 – 220. 3. Гаврилова Т.А. Базы знаний интеллектуальных систем / Т.А. Гаврилова, В.Ф. Хорошевский. – СПб.: Питер, 2001. – 384 с. 4. Литвин В.В. Базы знаний интеллектуальных систем поддержки принятия решений / В.В. Литвин. – Львів: Видавництво Львівської політехніки, 2011. – 240 с. 5. Постаногов Д.Ю. К вопросу многоязычности систем инженерии знаний и их приложений / Д.Ю. Постаногов, И.В. Совпель //

*Искусственный интеллект. – 2006. – Вып. 3. – С. 474 – 479.* 6. Miller G.A. WORDNET: A lexical database for English / G.A. Miller // *Communications of ACM.* – 1995. – N 11. – P. 39 – 41. 7. Гладун А.Я. Формирование тезауруса предметной области как средства моделирования информационных потребностей пользователя при поиске в Интернете / А.Я. Гладун, Ю.В. Рогушина // *Вестник компьютерных и информационных технологий.* – 2007. – № 1. – С. 26 – 33. 8. Литвин В.В. Оцінка новизни знань під час автоматичної розбудови онтологій / В.В. Литвин, А.С. Мельник, В.Я. Крайовський // *Інформаційні системи та мережі. Вісник НУ “Львівська політехніка”.* – 2011. – № 699. – С. 343 – 353. 9. *Інтелектуальні системи, базовані на онтологіях: монографія / Д.Г. Досин, В.В. Литвин, Ю.В. Нікольський, В.В. Пасічник.* – Львів: “Цивілізація”, 2009. – 414 с. 10. Литвин В.В. Моделювання плану поведінки інтелектуального агента на основі мереж Петрі та онтологічного підходу / В.В. Литвин // *Інформаційні системи та мережі: Вісник Нац. ун-ту “Львівська політехніка”.* – 2009. – № 653. – С. 170 – 175. 11. Рассел С. *Искусственный интеллект / С. Рассел, П. Норвиг.* – М.; СПб.; К.: Вильямс, 2006. – 1408 с. 12. Литвин В.В. Інтелектуальні агенти пошуку релевантних прецедентів на основі адаптивних онтологій / В.В. Литвин // *Математичні машини і системи.* – 2011. – № 3. – С. 66 – 72. 13. Литвин В.В. Моделювання інтелектуальних систем підтримки прийняття рішень з використанням онтологічного підходу / В.В. Литвин // *Радіоелектроніка, інформатика, управління / Запорізький національний технічний університет.* – 2011. – № 2 (25). – С. 93 – 101. 14. Новиков Ф.А. *Дискретная математика для программистов / Новиков Ф.А.* – СПб.: Питер, 2004. – [2-е изд.]. – 364 с.