

**Олег Сербін,**

старш. наук. співробітник НБУВ,  
канд. іст. наук

## **Систематизація бібліографічної інформації засобами концептуальних моделей знань**

У статті представлено концептуальні моделі знань як механізми систематизації бібліографічної інформації. Окреслено види та характеристики мов представлення онтологій. Подано детальний аналіз моделі структури опису даних – частини технології семантичної павутини та наведено приклади використання даної моделі для вираження триплетної організації інформації.

**К л ю ч о в і с л о в а:** триплет, онтологія, інформація, модель, веб-дані, система представлення знань, класифікація, база даних.

The publication presents the conceptual model of knowledge that is systematic mechanisms bibliographic information. Presents the types and characteristics of languages for representing ontologies. The detailed analysis of the structure model structure – data description of semantic web technologies and examples of using this model for expressing the triplet of information

**K e y w o r d s:** triplet, ontology, information, model, data, classification, database.

**І**нформація, з якою сучасний споживач має справу, досить багатогранна і різномірна. Коли йдеться про розуміння значення інформації, то мається на увазі потреба у представленні інформації таким чином, щоб показати її значущість в змістовному аспекті. Не є винятком і процес бібліографування, який відображає підготовку бібліографічної інформації. З огляду на те, що бібліографічна інформація – це інформація про документи, яка необхідна для їх ідентифікації та використання, можна виокремити ще два суміжних процеси, що впливають з наведеного вище – процеси ідентифікації та використання. Ідентифікуючи документ, ми маємо змогу його використовувати. Тому процес ідентифікації несе в собі набагато більше значення, ніж просто виявлення ідентичності певного документа. На нашу думку, процес ідентифікації включає в себе цілу низку процесів, які і розкривають зміст ідентифікації по суті. Більше того, щоб адекватно відобразити документ, щоб мати змогу його знайти і використати, вкрай важливо представити його таким чином, щоб інформація про нього була ідентифікаційною в будь-яких базах даних, базах знань.

Звичайно, представлення інформації в базах знань цілком можливе і щодо потреби систематизації галузевої інформації в рамках певного

розділу певного знаннєвого апарату. Тут питання в іншому: яким чином різні бази знань примусити взаємодіяти, а головне, розуміти одна одну? Коли йдеться про інтеперабельність, завжди розуміється уніфікація. Потрібно так уніфікувати інформаційні дані, щоб стало можливим представити та інтегрувати інформаційні ресурси незалежно від конкретної галузі знання чи загального змісту конкретного документа. В такому випадку постає потреба у створенні таких концептуальних моделей, механізмів організації, що дозволили б побудувати систему, в якій інформація пов'язувалася б поелементно, де заздалегідь були б прописані зв'язки між поняттями, значеннями, дескрипторами. Звичайно, оперативне опрацювання можливе тільки за допомогою нових сучасних технологій, сучасних моделей представлення та інтеграції інформаційних ресурсів, які активно розвиваються і впроваджуються в практику. Найважливішим елементом сучасних інформаційних технологій є онтології, які дають змогу здійснювати автоматизовану обробку семантики інформації, наданої через Інтернет, з метою її ефективного використання (представлення, перетворення, пошуку). Відповідний принцип обробки даних Інтернету ґрунтується на представленні останнього як гло-

бальної бази знань і орієнтованого не на осмислення інформації людиною [45], а на забезпечення семантичної інтероперабельності інформаційних ресурсів, тобто на автоматизовану інтерпретацію і обробку інформації. Цим і зумовлена актуальність заявленої теми. Об'єктом публікації є концептуальні моделі знань як механізми систематизації бібліографічної інформації. Предметом – процес систематизації бібліографічної інформації засобами концептуальних моделей знань. Метою публікації є визначення ролі та місця концептуальних моделей знань у процесі систематизації бібліографічної інформації.

Онтологія – термін, запозичений з філософії. Він означає науку, що описує форми буття і те, як вони співвідносяться між собою. Веб-онтологія може включати описи класів, властивостей та їх приклади. Онтології є новими інтелектуальними засобами пошуку ресурсів у мережі Інтернету, новими методами представлення та обробки знань і запитів. Вони здатні точно і ефективно описувати семантику даних для певної предметної області і вирішувати проблему несумісності і суперечності понять [3, с. 25–29]. У інформації онтологія розглядається як спроба усеосяжної і детальної формалізації деякої галузі знань за допомогою концептуальної схеми. Зазвичай така схема складається із структури даних, що містить усі релевантні класи об'єктів, їх зв'язки і правила (теореми, обмеження), прийняті в цій області. Онтології мають власні засоби обробки (логічного висновку), що відповідають завданням семантичної обробки інформації. Так, завдяки онтологіям при зверненні до пошукової системи користувач матиме можливість отримувати у відповідь ресурси, семантично релевантні запиту. Тому онтології набули значного поширення у вирішенні проблем представлення знань та інженерії знань, семантичної інтеграції інформаційних ресурсів, інформаційного пошуку тощо.

На сьогодні відомо декілька підходів до визначення поняття «онтологія», але загальноприйнятого визначення досі немає, оскільки залежно від конкретного завдання цей термін зручно інтерпретувати по-різному: від неформальних визначень до описів онтологій у поняттях і конструкціях логіки та математики. Ми тлумачитимемо цей термін як формальну специфікацію концептуалізації, що систематизується, має місце в деякому контексті предметної області і може виступати змістовною складовою процесу бібліографування. При цьому під концептуалізацією розумітимемо, окрім збирання

понять, також усю інформацію стосовно понять – властивості, стосунки, обмеження, аксіоми і твердження про поняття, необхідні для опису і вирішення завдань в обраній предметній області. Неформально онтологія складається з термінів і правил використання цих термінів, що обмежують їх значення у рамках конкретної області. На формальному ж рівні, онтологія – це система, що складається з набору понять і набору тверджень про ці поняття, на основі яких можна будувати класи, об'єкти, стосунки, функції, теорії.

Оскільки в кожній області можуть існувати різні розуміння одних і тих же термінів, онтологія виступає певною «угодою» щодо значення термінів і є посередником між людино- і машинно-орієнтованим рівнем представлення інформації. Онтологія існує у рамках домовленостей між членами співтовариства, наприклад, між користувачами деякої інформаційної системи. Концептуальне моделювання націлено на вирішення питань про те, як декларативним чином, що допускає повторне використання, описати предметну область, відповідні словники типів, як обмежити використання цих даних, з розумінням того, що може бути виведено з цього опису. З наведеного визначення також виходить, що окремими випадками онтологій є простий словник, тезаурус (у якому обмежено число відношень між термінами). Своєрідним представником певної онтології може бути і бібліографічна база даних з її змістовним наповненням та механізмами співвідношення понять. Фактично, онтологію можна застосовувати в якості компоненти баз знань, схеми об'єктів в об'єктно-орієнтованих системах, концептуальної схеми бази даних, структурованого глосарію інформаційних блоків, що взаємодіють, визначення класів для програмних систем. Онтології також дають змогу відповідним програмним засобам (інтелектуальним агентам<sup>1</sup>) автоматично (без участі людини) визначати сенс термінів, використаних при описі ресурсів, і зіставляти його з сенсом поставленого завдання.

Онтології мають дві особливості: вони можуть бути

<sup>1</sup> У комп'ютерній науці інтелектуальний (програмний) агент – це програма, що самостійно виконує завдання, вказане користувачем комп'ютера, протягом тривалих проміжків часу. Інтелектуальні агенти використовуються для сприяння операторові або для збирання інформації. Одним із прикладів завдань, що виконуються агентами, може бути завдання постійного пошуку й збирання необхідної інформації в Інтернеті [14].

множинними (складеними), в яких розрізняються представлення контексту одного і того ж домену<sup>2</sup>, а можуть ідентифікувати абстрактні рівні онтологій (бути рівнем вище інших онтологій). Стосовно другого випадку, то можливо ідентифікувати декілька рівнів абстракції, на кожному з яких можуть бути визначені онтології. Наприклад, в області кожної наукової дисципліни можна визначити онтології, а рівнем вище можна описати онтології наукових областей, що перебувають на стику окремих наукових дисциплін. Ще вище поставимо онтологію наукової дисципліни взагалі. Наступним рівнем абстракції можуть бути загальні категорії структур знань. Таке узагальнення приводить нас до необхідності детально розглядати умови створення онтологій, щоб організувати їх у бібліотеки онтологій.

Одним з основних аспектів організації онтологій є вибір відповідної мови специфікації (Ontology specification language). Мета таких мов – надати можливість вказувати додаткову машинозчитувану семантику ресурсів, для того щоб зробити машинне представлення даних більше схожим на дійсний стан речей у реальному світі, істотно підвищити виразні можливості концептуального моделювання веб-даних.

Сьогодні існує велика кількість мов представлення онтологій. Їх можна розподілити на певні групи: традиційні мови специфікації онтологій (KIF, Ontolingua, CycL); мови, засновані на дескриптивних логіках (LOOM); мови, засновані на фреймах<sup>3</sup> (OKBC, OCML, F-logic); мови, засновані на Web-стандартах (XOL, SHOE, RDF, DAML, OIL, OWL).

Досить часто для представлення онтологій використовується мова вказаної категорії KIF (Knowledge Interchange Format), розроблена на початку 90-х рр. в Лабораторії систем знань (KSL) Стенфордського університету. Спочатку вона розроблялася як формальна мова для забезпечення обміну знаннями між різними системами, заснованими на знаннях. Розробники анонсували [21], що даний формат не планується для використання представлення знань тільки в межах однієї програми, використання даного формату було націлено на взаємодію людини і

комп'ютера. Отже, дана мова фактично є посередником між програмними агентами.

На основі розширення мови KIF в тій же лабораторії було створено дослідницьку систему Ontolingua, що підтримує формування і представлення онтологій у певному канонічному форматі, завдяки чому забезпечується їх спільне використання та інтеграція у середовища різних систем, для здійснення певних операцій з ними. Онтологію, задану в класичному форматі, можна легко екстраполювати у різні системи, що використовують різний синтаксис для представлення знань і мають різні можливості міркувань. Система Ontolingua реалізована на мові Common Lisp, вона призначена для підтримки формальної специфікації завдань користувача на основі бібліотеки формальних описів фрагментів завдань, моделей і понять, а також для ведення самої бібліотеки фрагментів. Ontolingua включає програму синтаксичного аналізу, утиліту для перевірки правильності перехресних посилань, інструментів-перекладачів з мови системи на мови інших поширених систем представлення та генератор звітів у формі HTML. Як зазначалося вище, в основі мови формальних описів у системі Ontolingua лежить мова, розроблена в KSL і запропонована в якості стандарту міжмашинного обміну знаннями. Мова KIF є мовою числення предикатів другого порядку для представлення непроцедурних знань, доповнена засобами представлення процедурних знань у вигляді умовних правил переписування виразів. Користувач системи може бути видаленим і взаємодіяти з системою через Інтернет. Мета користувача – формалізувати завдання (чи модель) зі своєї прикладної області для наступних обчислень на спеціалізованих серверах математичної обробки. У бібліотеці Ontolingua зберігаються формальні описи фрагментів завдань користувача, кожне з яких, з математичної точки зору, є набором імен змінних, набором імен предикатів і функцій та набором аксіом або рівнянь, тобто те, що в математичній логіці називається теорією. Як правило, ці теорії не повні, вони описують лише те, що відомо про цей фрагмент. Користувач формальний опис свого завдання вибудовує з фрагментів, що зберігаються в бібліотеці Ontolingua, суміщаючи їх у різний спосіб і надаючи значення певним змінним. У результаті формується формальний опис завдання користувача, яке, з погляду математичної логіки, також представляється у вигляді теорії, тобто у вигляді набору змінних завдань, імен предикатів, імен функцій та набору аксіом і

<sup>2</sup> Домен (domain) – це область простору ієрархічних імен глобальної мережі Інтернету, яка обслуговується набором серверів доменних імен і централізовано адмініструється. Домен ідентифікується ім'ям домену [4].

<sup>3</sup> Фрейм (frame) – це окремий, закінчений HTML-документ, який разом з іншими HTML-документами може бути відображений у вікні веб-браузера [15].

рівнянь завдання. Завдання може полягати у визначенні значень деяких змінних чи області істинності деякої формули або у перевірці істинності деякої формули в побудованій теорії для цього завдання. Далі формальний опис завдання користувача може передаватися у форматі мови KIF на інші сервери через Інтернет, де для отримання відповідей на питання формальний опис може бути перекладений внутрішніми мовами інших систем і оброблений спеціальними програмами (рішенням рівнянь, програмними засобами спеціалізованої системи; логічного висновку, програмними засобами автоматичного доведення теорем; імітаційного моделювання тощо).

СусL – гібридна мова, в якій об'єднані властивості фреймів і логіки предикатів. Синтаксис мови СусL схожий на синтаксис мови Lisp [40]. СусL розрізняє такі сутності, як екземпляри, класи, предикати і функції. Словник СусL складається з термів<sup>4</sup>. Безліч термів можна розділити на константи, неатомарні терми і змінні. Терми використовуються при складанні значущих виразів СусL, з яких формуються судження. З суджень складається база знань. Дана мова – мова опису онтології Сус – проект зі створення об'ємної онтологічної бази знань, що дає змогу програмам вирішувати складні завдання в області штучного інтелекту на основі логічного висновку і залучення здорового глузду. Проект започаткував Дуглас Ленат у 1984 р. в Microelectronics and Computer Technology Corporation [24]. Назва «Сус» (утворена від «encyclopedia» і читається «цик»). База знань є власністю компанії, проте невелика частина бази, призначена для встановлення загального словника для програм автоматичного міркування, була випущена як OpenСус під відкритою ліцензією. База знань (англ. Knowledge Base або KB) містить більше мільйона, занесених туди людьми, тверджень, правил, загальноживаних ідей. Вони формулюються на мові СусL, яка заснована на численних предикатах і має схожий з ліспом<sup>5</sup> синтаксис. Велика частина роботи в проекті Сус ще і сьогодні пов'язана з інженерією знань – опис фактів про навколишній світ вручну та реалізація ефективних механізмів логічного висновку на основі цих

знань. Проте фахівці працюють над тим, щоб дати можливість системі Сус самостійно спілкуватися з користувачами на природній мові, а також, щоб прискорити процес поповнення бази за допомогою машинного навчання [40].

LOOM – це мова і середовище для побудови інтелектуальних доповнень [23]. Основою, ядром даної мови є система представлення знань, яка використовується для побудови дедуктивних висновків на основі декларативних знань. Останні складаються з визначень, правил, фактів і правил апіорі [29]. Дедуктивним рушієм виступають: використання прямих ланцюжків логічного висновку, семантична уніфікація, об'єктно-орієнтовані технології підтримки валідності. LOOM розроблена дослідниками з групи Artificial Intelligence Research Group [17] університету Південної Каліфорнії (University of Southern California's Information Sciences Institute), вона поширюється за відкритою ліцензією, але як інтелектуальна власність належить університету Південної Каліфорнії і не є загальнодоступною. Мета проекту LOOM – розроблення і впровадження засобів для представлення знань в області штучного інтелекту [8].

Open Knowledge Base Connectivity (OKBC) є інтерфейсом програмування доповнень для доступу до баз знань, що зберігаються в системах представлення знань. OKBC розробляється під егідою Управління перспективних дослідницьких проектів (DARPA) Пентагона для застосування в якості первинного протоколу для інтеграції різних компонентів технології. Цей протокол розроблений у Стенфордському університеті і є подальшим розвитком Generic Frame Protocol (GFP). OKBC забезпечує рівномірність моделі бази знань на основі загальної концепції класів, окремих осіб, слотів, наслідків (inheritance). Тобто, в рамках OKBC для знаннево-орієнтованих систем розроблена уніфікована модель системи представлення знань. Ця модель ґрунтується на теорії фреймів і використовує такі поняття, як «концептуалізація класів», «об'єкти», «слоти», «фасети», «спадкоємство» для подання знань про предметну область, що дає змогу створювати різні знаннево-орієнтовані додатки з високим рівнем інтероперабельності [27]. У рамках OKBC враховано багато специфічних особливостей систем представлення знань, розроблені у вигляді програмних компонентів редактори фреймів, візуалізація вмісту онтологій, інструменти аналізу і логічного висновку. Все це може задіюватися при розробленні специфічного програмного інструментарію, який

<sup>4</sup> Терм – вираження формальної мови (системи), формальне ім'я об'єкта або ім'я форми [33, р. 219].

<sup>5</sup> Мається на увазі Locator/Identifier Separation Protocol (LISP) – мова програмування загального призначення з підтримкою парадигм функціонального та процедурного програмування. Вихідна інформація записується у вигляді списків [19, р. 3–5].

використовує базу знань. ОКВС у мовах програмування визначається з позиції незалежності і в існуючих реалізаціях Common Lisp, Java та С. Протокол підтримує мережевий, а також прямий доступ до баз знань. ОКВС складається з безлічі операцій, які забезпечують загальний інтерфейс для різних баз знань [7]. Цей інтерфейс ізолює додатки від багатьох особливостей конкретної системи представлення знань і уможлиблює розроблення інструментів (наприклад, графічних браузерів, редакторів, засобів аналізу та виведення). ОКВС успішно використовується в декількох поточних проєктах в НДІ та Стенфордського університету.

OSML – мова моделювання підтримує формування моделей знань за допомогою декількох типів конструкцій [25]. Це дає змогу специфікувати і вводити в дію функції, відношення, класи, екземпляри і правила [28]. Вона також включає механізми для визначення онтологій, вирішення проблем методів і основних технологій, розроблених в області моделювання знань. OSML використовується у низці проєктів, які реалізуються КМІ (Knowledge Media Institute, Англія) [20]. За допомогою цієї мови забезпечується моделювання для підтримки додатків у областях, заснованих на системі знань, зокрема таких, як управління знаннями, онтологія розвитку, електронна торгівля. OSML підтримується великою бібліотекою моделей, що повторно використовуються. Це дає змогу забезпечувати корисний ресурс для моделювання систем знань.

F-Logic – онтологічна мова, яка ґрунтується на логіках першого порядку, проте класи і властивості в ній представлені як терміни, а не як предикати [8]. Вона створювалася для здійснення взаємодії між онтологіями, побудованими на основі предикатів, і онтологіями, побудованими на основі F-Logic. Її творці визначили інтуїтивні транслятори для перетворення знань з предикативних онтологій у F-Logic онтології і показали, що такий переклад зберігає логічні зв'язки для великої кількості онтологічних мов [26], у т. ч. і для багатьох OWL DL, про що йтиметься нижче. F-Logic може застосовуватися для метамодельювання розширень Description Logics <sup>6</sup>.

<sup>6</sup> Описові логіки (англ. Description logics, іноді їх ще називають дескрипційними логіками) – сімейство мов представлення знань, що дають змогу описувати поняття предметної області в недвозначному, формалізованому вигляді. Будь яка описова логіка є також логікою першого порядку, але не навпаки. Останнім часом зріс інтерес до описових логік через просування семантичної павутини [9].

XOL – мова для обміну онтологіями. Вона призначена для використання в якості мови-посередника для передачі онтологій між різними системами управління базами даних, онтологіями розвитку інструментів або прикладними програмами. XOL надає формат для обміну визначень онтологій серед безлічі заінтересованих сторін. Дефініції, призначені для кодування, включені до XOL як схеми інформації (мета-дані). Йдеться, насамперед, про визначення класу об'єкта з бази даних, і нонсхеми інформації, такі як визначення об'єктів від об'єкта баз даних. Синтаксис XOL заснований на XML, який є мовою для створення документів для WWW <sup>7</sup>. XML-синтаксис обрано тому, що його синтаксис і параметри досить прості для розуміння людиною. Семантика XOL заснована на ОКВС-Lite, яка є спрощеною формою моделі знань для ОКВС (Open Knowledge Base Connectivity). ОКВС – це API (Application program interface) для доступу до системи знань. Його модель знань підтримує можливості, які найчастіше існують у системах представлення знань, об'єктних базах і реляційних базах даних. Наприклад, група з розроблення наукової бази даних може використовувати систему керування базами даних (СКБД) Oracle для створення реляційної бази даних. Ставлячи перед собою мету перевести СКБД схему з SQL <sup>8</sup> у XOL, а потім опублікувати отриманий файл на WWW для довідки користувачам бази даних або інших груп, можна перетворити схеми, використовуючи існуючий SQL-на-XOL перекладач. XOL схожий на інші, розроблені раніше, мови представлення онтологій. Зокрема, історія його починається із згаданої вище Ontolingua. Насправді, XOL відрізняється від Ontolingua наявним у його основі XML-синтаксисом, а не Lisp-синтаксисом, хоча заради справедливості варто зазначити, що семантика ОКВС-Lite дуже схожа на семантику Ontolingua.

<sup>7</sup> Всесвітнє павутиння (англ. World Wide Web) – розподілена система, що надає доступ до пов'язаних між собою документів, розташованих на різних комп'ютерах, підключених до Інтернету. Для позначення Всесвітньої павутини також використовують слово «веб» (англ. web), «W3» і «WWW» [47].

<sup>8</sup> SQL (англ. Structured query language – мова структурованих запитів) – декларативна мова програмування для взаємодії користувача з базами даних, що застосовується для формування запитів, оновлення і керування реляційними БД, створення схеми бази даних і її модифікації, система контролю за доступом до бази даних [36].

SHOE – маленьке розширення до HTML, яке дає змогу авторам веб-сторінки анотувати свої веб-документи за допомогою машиноорієнтованого знання. SHOE робить можливим реальне інтелектуальне програмне забезпечення агента в мережі, HTML ніколи не призначався для суто комп'ютерного використання. Його призначення – показати дані для людей, щоб читати. «Знання» про веб-сторінку представлено зручною для читання мовою (зазвичай англійською), а також графікою і структурним способом, що дає змогу людині сприймати знання візуально. На жаль, розумні програмні агенти не є людьми. Навіть з сучасною технологією природної мови змусити комп'ютер читати і розуміти веб-документи дуже важко. Це, у свою чергу, спричиняє труднощі щодо створення розумного агента, який би міг вільно блукати мережею, досліджувати і розуміти зміст веб-сторінок. SHOE нівелює цю проблему, надаючи можливість представляти веб-сторінки засобами знань, які розуміє і знає агент.

RDF – спосіб представлення знань у децентралізованому світі. Він є технологією, яка дає змогу комп'ютерним програмам користуватися усією структурованою інформацією, розподіленою по вузлах Веба [6]. RDF не є XML-форматом у буквальному розумінні – це більш ніж просто метадані. Фактично це універсальний спосіб представлення будь-якого цілісного знання засобами невеличких окремих частинок. Він задає певні правила стосовно семантики, тобто сенсу цих частинок. Ідея полягає в тому, щоб одним простим способом можна було описати будь-який факт, причому в такому структурованому вигляді, щоб його могли обробляти комп'ютерні програми. RDF призначений для представлення знань у розподіленому світі. Іншими словами, для RDF особливо важлива змістовність, все з чим він працює повинно мати і має певну змістовність, тому і посилання робиться на якийсь конкретний об'єкт, або на абстрактне поняття, або на той чи інший факт [32]. Стандарти, засновані на RDF, описують логічні висновки, що пов'язують ці факти, і вказують, як можна знайти самі факти у величезній базі даних усіх знань, представлених у RDF. Як зазначалося вище, RDF підходить для роботи з розподіленими знаннями тому, що доповнення можуть збирати разом RDF-файли, розміщені в Інтернеті різними людьми, і з легкістю дізнаватися із зібраного документа навіть про те нове, що не було представлено у жодній з його частин до цього. У RDF передбачається два процеси, завдяки

яким все це здійснюється. По-перше, об'єднуються документи, що використовують спільні мови, по-друге, допускається використовувати будь-які мови у кожному з документів. Вищевикладене характеризує RDF як доволі гнучку технологію, що є дуже важливим і, більше того, необхідним у процесі уніфікації представлення знань. Цим зумовлена і доволі широка панорама застосування RDF, перелік можливостей застосування якої представлено у списку розсилки W3C<sup>9</sup> Semantic Web [34]: об'єднання даних з різних джерел, не створюючи спеціалізованих програм; можливість надання доступу іншим користувачам до особистих даних; децентралізація даних до такої міри, щоб ними користувався не тільки хтось один; відтворення специфічних процесів з великими обсягами даних – введення, виведення, перегляд, аналіз, здійснення пошуку, що дає змогу створювати (або використовувати готовий) універсальний інструмент, засобами якого можна було б усе це робити, дотримуючись моделі даних RDF (такої, яка має перевагу в тому, що вона не прив'язана до закритих технологій зберігання і представлення даних, на відміну від діалектів СКБД).

Будь-яке вираження RDF ґрунтується на колекції так званих триплетів, кожен з яких складається з суб'єкта, предиката і об'єкта. Кожен триплет є свідченням зв'язку між поняттями (суб'єкт і об'єкт) [5], позначеними як модулі, котрі цей зв'язок сполучає. Варто наголосити, такий зв'язок має певний напрям і, що особливо важливо, спрямування завжди відбувається від суб'єкта до об'єкта. Таким чином, RDF-граф утворює семантичну мережу. Значенням RDF-графа є логічне «І» усіх тверджень-триплетів, що формують цей граф. Варто принагідно зазначити, що граф – це сукупність вузлів і ребер, що сполучають ці вузли. Тобто це система, яка інтуїтивно може бути розглянута як безліч кіл і безліч ліній, що сполучають їх. Кола називаються вершинами графа, лінії зі стрілками – дугами, без стрілок – ребрами. Граф, в

<sup>9</sup> Консорціум Всесвітньої павутини (англ. World Wide Web Consortium, W3C) – головна міжнародна інституція, яка розробляє й упроваджує технологічні стандарти для Інтернету. Заснована 1994 р. Тімом Бернерсом-Лі – винахідником Всесвітньої павутини, котрий її і очолює. Прикладами стандартів Консорціуму є HTML, XML, CSS, SVG, RSS та WCAG. Юридично W3C не є міжнародною організацією стандартизації, проте її стандарти отримують загальне визнання і вони вільні від будь-яких платних запатентованих технологій [41].

якому напрям ліній не виокремлюється (усі лінії є ребрами), називається неорієнтованим, а граф, в якому напрям ліній принципівий (лінії є дугами) називається орієнтованим [1, с. 47–51]. В чистому програмуванні виділяють також об'єктні графи, які забезпечують простий спосіб обліку взаємних зв'язків у безлічі об'єктів, і не обов'язково, щоб ці зв'язки точно проектувалися в класичні зв'язки об'єктно-орієнтованого програмування (такі як відношення головного і підлеглого), хоча вони моделюють цю парадигму досить добре [16]. Кожному об'єкту в об'єктному графові присвоюється унікальне числове значення. Ці числові значення, що приписуються членам в об'єктному графі, довільні і не мають ніякого сенсу поза графом. Після присвоєння усім об'єктам числового значення, об'єктний граф може почати запис безлічі залежностей кожного об'єкта.

Таким чином, RDF формує модель для анотації ресурсів мережі за допомогою XML-синтаксису і триплетної моделі, яка має наочне графічне представлення (орієнтований розмічений граф). Пояснимо докладніше. Для прикладу представимо такі речення:

*Класифікація двокрапкою розроблена Шіалі Ранганатаном.*

*Класифікація двокрапкою є фасетною класифікацією.*

У даному випадку ми маємо два триплети, а саме: «Класифікація двокрапкою» – об'єкт, «розроблена» і «є» – предикати, «Шіалі Ранганатан» і «фасетною класифікацією» – об'єкти в даних триплетах.

Розуміючи, що завдання будь-якого триплету – це представлення інформації у такому вигляді, щоб завжди зберігалася функціональна тотожність між трьома складовими триплетної моделі, тобто, щоб при потребі об'єкт міг виступати у ролі предиката або суб'єкта в рамках іншого триплету, і так по кожному з вищеперелічених елементів. Отже, з даних двох триплетів впливає таке речення:

*Шіалі Ранганатан розробив фасетну класифікацію.*

Таким чином, два об'єкти перших двох триплетів «Шіалі Ранганатан» та «фасетна класифікація» через предикати «є розробником» утворили новий триплет «Шіалі Ранганатан» «Розробник» «Фасетна класифікація». У новоутвореному триплеті «Шіалі Ранганатан» вже є не об'єктом, а суб'єктом, що і підтверджує універсальність триплетної організації даних.

На рис. 1 схематично представлено взаємозв'язок триплетів:



Рис. 1. Взаємозв'язок триплетів

З наведеного рисунка видно, як утворюється новий триплет. Грамотна триплетна модель якраз таки націлена на формування нескінченної кількості зв'язків для представлення інформації різноаспектно і функціонально. Якщо ж далі рухатись у цьому напрямі, то можна говорити, що «особливістю фасетної класифікації є застосування нової форми організації даних – фасетів». Якщо спростити до триплетного бачення, то «фасетна класифікація складена з фасетів». Звідси впливає і те, що класифікація двокрапкою складається з фасетів, і те що Шіалі Ранганатан розробив фасети. І так можна розвивати триплетну модель для представлення всіх граней, які розкривають змістовність питання щодо «Класифікації двокрапкою» Ш. А. Ранганатана.

Кожне речення, таким чином, має підмет (суб'єкт), присудок (предикат) і доповнення (об'єкт). Підмет представляє ресурс, присудок – властивість, а доповнення – значення властивості. Перетворивши ці речення в щось схоже на трійку RDF, отримаємо такий результат:

«класифікація двокрапкою» «розробити» «Шіалі Ранганатан»;

«класифікація двокрапкою» «є» «фасетна класифікація».

Говорячи про RDF, нам потрібно представити ресурс, який ототожнюється з елементом триплету. Значимо, що ресурс може бути про що завгодно. Важливо, щоб можна було посилатися на нього за допомогою Уніфікованого ідентифікатора ресурсу<sup>10</sup> (Uniform Resource Identifier) або URI. Особливим випадком URI є Уніфікований покажчик ресурсу<sup>11</sup> (Uniform Resource Locator) або URL, який використовується в

<sup>10</sup> Це символічний рядок, що дає змогу ідентифікувати який-небудь ресурс : документ, зображення, файл, службу, ящик електронної пошти і т. д. URI надає простий і розширений спосіб ідентифікації ресурсів [42].

<sup>11</sup> URL включає: метод доступу до ресурсу, тобто протокол доступу (http, gopher, WAIS, ftp, file, telnet та ін.); мережеву адресу ресурсу (ім'я хост-машини і домену); повний шлях до файла на сервері [43].

роботі з Інтернет і може мати, наприклад, такий вигляд: <http://www.nbu.gov.ua> або [library@nbuv.org.ua](mailto:library@nbuv.org.ua). URI може бути загальнішим, і тоді вже йдеться про Уніфіковане ім'я ресурсу<sup>12</sup> (Uniform Resource Name) або URN замість URL. Це означає, що URL використовується для визначення ресурсів, таких як веб-сторінки і поштові скриньки електронної пошти – ресурси, для яких URL може надати необхідну і достатню для пошуку інформацію та URN для визначення інших ресурсів. Найважливіше – це можливість визначити ресурс за допомогою деякого URI, а форма його подання у даному випадку не важлива.

Враховуючи все вищесказане, представимо триплети за допомогою RDF з використанням певних URI. Для цього наведемо два приклади з отриманим вище реченням «Ранганатан – розробник фасетної класифікації». Перший приклад характеризуватиметься використанням концептів<sup>13</sup> у вигляді ресурсів, другий – використанням концептів у вигляді літералів<sup>14</sup>.

*Приклад 1.* «Ранганатан – розробник фасетної класифікації» (з концептами у вигляді ресурсів).

Всі властивості для прикладу візьмемо з базових схем (rdf і dc). Запишемо приклад у синтаксисі RDF/XML і у вигляді таблиці.

```
<rdf:RDF
  xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:dc="http://purl.org/dc/elements/1.1/">
  <rdf:Description rdf:About="http://serbin.org/people/ranganathan.html">
  </rdf:Description>
  <rdf:Description rdf:About="http://serbin.org/classifications/ranganathan.html">
  <rdf:Type rdf:Resource="http://serbin.org/principles/facet.html">
  <dc:Creator rdf:Resource="http://serbin.org/people/ranganathan.html">
  </rdf:Description>
</rdf:RDF>
```

<sup>12</sup> Власні імена URN покликані в майбутньому замінити локатори URL. Але імена URN, на відміну від URL, не включають вказівки щодо місцезнаходження та способу звернення до ресурсу [44].

<sup>13</sup> Концепт – (від лат. *conceptus* – думка – поняття), 1) смислове значення імені (знаку), тобто зміст поняття, обсяг якого є предмет (денотат) цього імені (напр., смислове значення імені Індекс – об'єкт бази даних, створений з метою підвищення ефективності виконання запитів) [18, р. 7–9].

<sup>14</sup> Літерал (англ. *literal* – константа) – запис у початковому коді комп'ютерної програми, що є фіксованим значенням [37].

У приведеній нижче таблиці представлено триплети відповідно «суб'єкт» – «предикат» – «об'єкт»:

Суб'єкт	Предикат	Об'єкт
<a href="http://serbin.org/classifications/ranganathan.html">http://serbin.org/classifications/ranganathan.html</a>	dc:Creator	<a href="http://serbin.org/people/ranganathan.html">http://serbin.org/people/ranganathan.html</a>
<a href="http://serbin.org/classifications/ranganathan.html">http://serbin.org/classifications/ranganathan.html</a>	rdf:Type	<a href="http://serbin.org/principles/facet.html">http://serbin.org/principles/facet.html</a>

Таким чином, якщо:

суб'єкт «<http://serbin.org/classifications/ranganathan.html>» співвідноситься з об'єктом «<http://serbin.org/people/ranganathan.html>» через предикат dc:Creator

і:

суб'єкт «<http://serbin.org/classifications/ranganathan.html>» співвідноситься з об'єктом «<http://serbin.org/principles/facet.html>» через предикат rdf:Type,

то правильним буде таке твердження:

суб'єкт «<http://serbin.org/people/ranganathan.html>» співвідноситься з об'єктом «<http://serbin.org/principles/facet.html>» через предикат «<http://serbin.org/classifications/ranganathan.html>»

Отже, зрозумілим є те, що приклад у неявному вигляді містить:

```
<rdf:Description rdf:About="http://serbin.org/principles/facet.html">
  <dc:creator rdf:Resource="http://serbin.org/people/ranganathan.html"/>
</rdf:Description>
```

*Приклад 2.* «Ранганатан – розробник фасетної класифікації» (з концептами у вигляді літералів + описуємо одну веб-сторінку).

Припустимо, що ми маємо веб-сторінку про історію класифікацій. На даній сторінці містяться розділи, присвячені особистості Ранганатана, його класифікації та фасетним класифікаціям загалом. Введемо ідентифікатори літералів для всіх трьох концептів даного прикладу. Запишемо все у синтаксисі RDF/XML.

```
<rdf:RDF
  xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:dc="http://purl.org/dc/elements/1.1/">
  <rdf:Description
    rdf:About="http://serbin.org/historyOfClassifications.
```



```

html#Ranganatan»>
  <rdf:ID>Панганатан</rdf:ID>
  </rdf:Description>
  <rdf:Description
rdf:About=«http://serbin.org/historyOfClassifications
.html#Colon»>
  <rdf:ID>Класифікація двокрапкою</rdf:ID>
  <rdf:Type rdf:ID=«Фасетна класифікація»>
  </rdf:Description>
  <rdf:Description
rdf:About=«http://serbin.org/historyOfClassifications
.html#Facet»>
  <rdf:ID> Фасетна класифікація </rdf:ID>
  <dc:Creator rdf:ID=«Панганатан»>
  </rdf:Description>
</rdf:RDF>

```

Таким чином, якщо:

суб'єкт «Класифікація двокрапкою» співвідноситься з об'єктом «Панганатан» через предикат `dc:Creator`

і:

суб'єкт «Класифікація двокрапкою» співвідноситься з об'єктом «Фасетна класифікація» через предикат `rdf:Type`,

то правильним буде таке твердження:

суб'єкт «Панганатан» співвідноситься з об'єктом «Фасетна класифікація» через предикат «Класифікація двокрапкою».

Отже, зрозумілим є те, що приклад у неявному вигляді містить:

```

<rdf:Description rdf:About=«http://serbin.org/principles/facet.html»>
<dc:creator rdf:ID=«Панганатан»>
</rdf:Description>

```

Фактично ми продемонстрували, як з двох відомих триплетів отримується третій неявний. Більше того, наведеними прикладами ми ще раз довели унікальність та особливість використання триплетного моделювання інформації, враховуючи, що в новому триплеті «Панганатан»–«Класифікація двокрапкою»–«Фасетна класифікація» суб'єктом виступає об'єкт «Панганатан» триплету «Класифікація двокрапкою»–«dc:Creator»–«Панганатан». Для наочності представимо тепер всі триплету як орієнтований граф (див. рис. 2).

Однак, слід зауважити, що RDF у чистому вигляді описує як вказувати інформацію про ресурс, але сама інформація не має значення. Для визначення типів елементів і їх відношень, таких як класи і підкласи, необхідно розширити словниковий запас [11]. Будь-які розширення словникового запасу фіксуються у мові представлення RDF

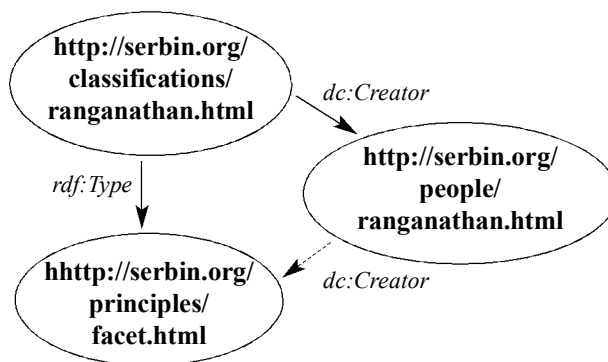


Рис. 2. Триплетна модель у графах

Schema (RDFS). Мова представлення RDFS має власну область імен, набір зумовлених елементів і атрибутів. RDFS – це стандарт ініціативи W3C для представлення онтологічних знань. RDFS є мовою опису за допомогою стандартизованого набору тегів у форматі RDF-триплетів (який і є основою RDFS), що призначена для вираження словників користувачів. Іншими словами, RDFS надає систему типів для RDF. Власне сама RDF як така не є осередком чи втіленням якихось механізмів [35, р. 12–18] ні для представлення властивостей, ні для подання відношень між цими властивостями та іншими ресурсами. Дану функцію сповна виконує саме RDFS, яка визначає поняття «клас», «властивість» і деякі інші, що дає змогу описувати класи, властивості та інші ресурси. Тобто, RDF моделі предметних областей описуються за допомогою ресурсів, властивостей та їх значень [10], а RDFS, у свою чергу, специфікує безліч всіляких допустимих схем даних. Певним недоліком RDFS є те, що за допомогою цієї мови не можна задати аксіоми і правила виводу, побудовані на аксіомах, тобто виразити аксіоматичні знання. Проте збільшення виразних здібностей інструменту RDFS можливе і надзвичайно корисне. Механізм розширення внутрішніх властивий RDFS можливий через уточнення або доповнення базових типів. Тому RDFS став фундаментом для багатших мов опису концепцій предметних областей, званих мовами опису онтологій предметних областей, про які йтиметься далі.

Одними з перших пропозицій з опису онтологій на основі RDFS були DAML (DARPA Agent Markup Language) [13] і OIL (Ontology Inference Layer) [38]. Ці стандарти специфікації і обміну онтологіями розроблялися для досягнення найкращих результатів щодо підтримки процесу обміну знаннями та інтеграції знань. DAML представляє

примітиви<sup>15</sup> для визначення перетинів, об'єднань, доповнень класів. У свою чергу, OIL заснований на згаданих вже описових логіках.

На базі DAML і OIL виникло спільне рішення – DAML+OIL – семантичної мови розмітки Web-ресурсів, яка розширює стандарти RDF і RDFS за рахунок повніших примітивів моделювання. В останню версію DAML OIL включений набір додаткових конструкцій для кращого створення онтологій і розмітки інформації. Це стало поштовхом до створення у рамках ініціативи Semantic Web<sup>16</sup> окремої групи з перегляду і стандартизації мови опису Web-онтологій (OWL – Web Ontology Language).

OWL – спроба зробити веб-ресурси доступнішими для автоматизованих процесів шляхом додавання інформації про ресурси, яка описує або забезпечує здійснення веб-контенту [22, р. 3–7]. Оскільки Семантична павутина розподілена, OWL покликає збирати інформацію з різних джерел. Це частково забезпечується можливістю онтологій бути пов'язаними, включаючи прямий імпорт інформації з інших онтологій. Застосування OWL припускає відкритість [12]. Тобто, описування ресурсів не обмежене єдиним файлом або темою. Нова інформація може бути суперечливою, але факти і наслідки можуть тільки додаватися, вони не можуть видалятися, бути проігнорованими. При потребі в OWL для її використання різними розробниками і користувачами, передбачено три різні діалекти: OWL Lite, OWL DL, OWL Full [31]. Зокрема, OWL Lite корисний для тих користувачів, котрі потребують оперування класифікаційною ієрархією. Пріоритетність для розробників використання OWL Lite продиктована можливістю швидкої міграції тезаурусів та інших таксономій. Разом з тим, для отримання максимальної виразності без втрати повноти обчислень (усі висновки гарантовано будуть обчислюваними) і розсудливих систем, що мають здатність розширюватись (всі обчислення завершаться в певний час), використовується OWL DL [30], який містить усі мовні конструкції OWL з обмеженнями на зразок розподілу типу. Тобто клас не може бути особистою властивістю, а

властивість не може бути індивідом або класом. Виходячи з того, що OWL DL – це OWL Description Logics, можна зробити висновок: даний діалект належить до сімейства мов представлення знань, які дають змогу описувати поняття предметної області в недвозначному, формалізованому вигляді. З іншого боку, OWL Full призначається для користувачів, які прагнуть отримати максимальну виразність і синтаксичну свободу RDF без обчислювальних гарантій [46]. OWL Full представляє онтології, які розширюють склад існуючого (RDF або OWL) словника. Але маловірогідно, що якесь розсудливе програмне забезпечення зможе забезпечити повну підтримку кожної особливості OWL Full. Кожен з цих діалектів – це фактично вдосконалення та розширення його простішого попередника як в аспекті можливостей вираження, так і в аспекті можливостей, що зумовлені висновками.

Формальна семантика OWL описує як отримати логічні наслідки, маючи таку онтологію, тобто отримати факти, які не представлені в онтології буквально, але виходять з її семантики. Ці наслідки можуть бути зосереджені на одному документі або безлічі розподілених документів, що комбінуються з використанням певних механізмів OWL.

Загалом відмінність між традиційними і веб-мовами специфікації онтологій полягає у виразних можливостях опису предметної області і деяких можливостях механізму логічного висновку для цих мов. Типові примітиви мов додатково включають: конструкції для агрегації, множинних ієрархій класів, правил виведення і аксіом; різні форми модуляризації для запису онтологій і взаємин між ними, можливість метаописування онтологій, що, в свою чергу, необхідно при встановленні відношень між різними видами онтологій.

На сьогоднішній день є закономірним використання концептуальних моделей знань у процесі систематизації (організації) інформаційних масивів. Будь-які технологічні можливості, що з'являються та розвиваються таким чином загальні підходи до бібліографування даних, необхідно не тільки знати, а й робити певні кроки щодо практичного використання результатів еволюційного розвитку Інтернет-технологій. За такої ситуації на перший план фактично виходить розуміння значення не тільки змісту, суті документа, а й форми уніфікованого представлення знань для забезпечення інтероперабельності баз даних. Навіть спосіб уніфікації інформації, у контексті сказаного вище, розглядається нами як наслідок бачення потреб тієї форми подання даних, котра як

<sup>15</sup> Примітиви – це заздалегідь визначені елементи, гранично прості одиниці [2].

<sup>16</sup> Семантична павутина – це надбудова над існуючою Всесвітньою павутиною, яка покликає зробити розміщену в ній інформацію зрозумілішою для комп'ютерів. Машинна обробка можлива в семантичній павутині завдяки її найважливішим характеристикам: використання URI, онтологій та семантичних мереж [39].

орієнтир встановлюється задля досягнення мети систематизації.

Отже, таке розмаїття веб-мов характеризує концептуальні моделі знань не з позиції розшарпаності та хаосу, а як такі, що націлені на представлення різних підходів і фактично механізмів вираження зв'язків між різнорідними поняттями. Більше того, використання онтологій (у рамках представленого вище термінологічного окреслення даного терміна) практично пояснює новатику підходів щодо систематизації інформації, бібліографічної у першу чергу, враховуючи тематичну спрямованість даного дослідження.

Проведене дослідження дає змогу зробити висновки щодо сьогодення і перспектив ситуації навколо організації інформації з урахуванням можливостей Інтернету і потреб користувачів-читачів. В першу чергу, можна припустити, що основа організації знань має і по суті, і по формі трилогічний характер. Основна мета представлення інформації у триплетний спосіб полягає в адекватному до потреб користувача поданні даних задля різностороннього використання певних термінів, понять, дескрипторів. У такий спосіб не тільки здійснюється релевантний пошук, але й відбувається розвиток (у процесі пошуку) системи, яка представляє ту чи іншу предметну область. Крім того, у процесі систематизації інформації засобами концептуальних моделей знань робиться певний внесок у вже існуючу систему і представляються абсолютно нові відомості про галузь, яку користувач спочатку вважав не основною, або ж такого, що не мала змістовного значення для його (користувача) роботи. Тому представлення бібліографічної інформації засобами концептуальних моделей знань – це результат еволюції механізмів організації таких онтологій, способів та форм їх представлення, а також копійки роботи щодо систематизації бібліографічних даних та класифікації інформації.

### Список використаних джерел

1. Бурков В. Н. Теория графов в управлении организационными системами / В. Н. Бурков, А. Ю. Заложнев, Д. А. Новиков. – М.: Синтег, 2001. – 124 с.
2. Вежбицка А. Из книги «Семантические примитивы». Введение // Семиотика / под ред. Ю. С. Степанова. – М., 1983. – С. 137–150.
3. Добров Б. В. Онтологии и тезаурусы: модели, инструменты, приложения / Б. В. Добров, В. В. Иванов, Н. В. Лукашевич, В. Д. Соловьев. – М.: Интернет-Университет Информационных Технологий; БИНОМ. Лаборатория знаний, 2009. – 173 с.
4. Домен (domain) [Електронний ресурс]. – Режим доступу : [mchost.com.ua/staty/domain.html](http://mchost.com.ua/staty/domain.html) – Назва з екрана.
5. Казекин М. М. История языков представления онтологий // Компьютерные инструменты в образовании. – 2008. – № 4. – С. 3–11.
6. Краткое введение в RDF [Електронний ресурс]. – Режим доступу : URL : <http://xmlhack.ru/texts/06/rdf-quickintro/rdf-quickintro.html>. – Название с экрана. – Дата просмотра : 5 октября 2010.
7. Несколько слов о протоколе Open Knowledge Base Connectivity (OKBC) и редакторе онтологий Protégé [Електронний ресурс]. – Режим доступу : URL : <http://shcherbak.net/2008/01/okbc/#more-27>. – Название с экрана. – Дата просмотра : 25 сентября 2010.
8. Никоненко А. А. Обзор баз знаний онтологического типа // Штучный интеллект. – 2009. – № 4. – С. 208–219.
9. Описові логіки [Електронний ресурс]. – Режим доступу : URL : [http://uk.wikipedia.org/wiki/Описова\\_логіка](http://uk.wikipedia.org/wiki/Описова_логіка). – Назва з екрана. – Дата перегляду : 22 вересня 2010.
10. Основы RDF [Електронний ресурс]. – Режим доступу : URL : <http://www.ibm.com/developerworks/ru/edu/x-ultimashup3/section3.html>. – Название с экрана. – Дата просмотра : 5 октября 2010.
11. Язык описания RDF-схемы (RDF-схема) [Електронний ресурс]. – Режим доступу : URL : <http://www.ibm.com/developerworks/ru/edu/x-ultimashup3/section5.html>. – Название с экрана. – Дата просмотра : 5 октября 2010.
12. A Quick Introduction to OWL Web Ontology Language [Virtual Resource]. – Access Mode : URL : <http://www.iro.umontreal.ca/~lapalme/ift6281/OWL/CostelloQuickIntroOwl.pdf>. – Title from Screen. – Date of Access : 24 September 2010.
13. About the DAML Language [Virtual Resource]. – Access Mode : URL : <http://www.daml.org/about.html>. – Title from Screen. – Date of Access : 12 September 2010.
14. Agents: Intelligent Assistants Working With You and For You [Virtual Resource]. – Access Mode : URL : <http://www.aaai.org/AITopics/pmwiki/pmwiki.php/AITopics/Agents>. – Title from Screen. – Date of Access : 12 September 2010.
15. Dan's Web Tips: Frames [Virtual Resource]. – Access Mode : URL : <http://webtips.dan.info/frames.html>. – Title from Screen. – Date of Access : 3 October 2010.
16. Intelligent Graph Visualizer [Virtual Resource]. – Access Mode : URL : <http://sourceforge.net/projects/igv-intelligent/>. – Title from Screen. – Date of Access : 11 October 2010.
17. Intelligent Systems Division. USC/Information Sciences Institute (ISI). [Virtual Resource]. – Access Mode : URL : <http://ai.isi.edu/>. – Title from Screen. – Date of Access : 20 September 2010.
18. Jerry A. Fodor. Concepts: Where Cognitive Science

- Went Wrong. – USA : Oxford University Press, 1998. – 192 p.
19. John McCarthy. LISP 1.5 programmer's manual. – Cambridge, Mass. : M.I.T., 1968. – 106 p.
20. KMi [Virtual Resource]. – Access Mode : URL : <http://kmi.open.ac.uk/> – Title from Screen. – Date of Access : 20 September 2010.
21. Knowledge Interchange Format (KIF) [Virtual Resource]. – Access Mode : URL : <http://www-ksl.stanford.edu/knowledge-sharing/kif/>. – Title from Screen. – Date of Access : 24 September 2010.
22. Lee W. Lacy. Owl: Representing Information Using The Web Ontology Language. – Canada. : Trafford, 2005. – 282 p.
23. Loom Project Home Page. Artificial Intelligence research group University of Southern California's Information Sciences Institute. [Virtual Resource]. – Access Mode : URL : <http://www.isi.edu/isd/LOOM/>. – Title from Screen. – Date of Access : 24 September 2010.
24. Microelectronics and Computer Technology Corporation [Virtual Resource]. – Access Mode : URL : <http://foldoc.org/Microelectronics+and+Computer+Technology+Corporation>. – Title from Screen. – Date of Access : 20 September 2010.
25. OCML Ontologies to XML Schema Lowering [Virtual Resource]. – Access Mode : URL : <http://cite-seerx.ist.psu.edu/viewdoc/download?doi=10.1.1.143.490&rep=rep1&type=pdf>. – Title from Screen. – Date of Access : 20 September 2010.
26. On the Relationship between Description Logic-based and F-Logic-based Ontologies / Jos de Bruijn, Stijn Heymans // Fundam. Inform. – 2008. – № 82(3). – P. 213–236.
27. Open Knowledge Base Connectivity Home Page [Virtual Resource]. – Access Mode : URL : <http://www.ai.sri.com/~okbc/>. – Title from Screen. – Date of Access : 24 September 2010.
28. Operational Conceptual Modelling Language [Virtual Resource]. – Access Mode : URL : <http://technologies.kmi.open.ac.uk/ocml/>. – Title from Screen. – Date of Access : 11 October 2010.
29. Overview: Loom Knowledge Representation System [Virtual Resource]. – Access Mode : URL : <http://www.isi.edu/isd/LOOM/#OVERVIEW>. – Title from Screen. – Date of Access : 24 September 2010.
30. OWL DL Semantics [Virtual Resource]. – Access Mode : URL : <http://www.obitko.com/tutorials/ontologies-semantic-web/owl-dl-semantics.html>. – Title from Screen. – Date of Access : 20 September 2010.
31. OWL Web Ontology Language. Guide [Virtual Resource]. – Access Mode : URL : <http://www.w3.org/TR/owl-guide/> – Title from Screen. – Date of Access : 11 October 2010.
32. RDF for dummies [Virtual Resource]. – Access Mode : URL : <http://habrahabr.ru/blogs/sw/17946/>. – Title from Screen. – Date of Access : 5 October 2010.
33. Schwartzman Steven. The words of mathematics: An etymological dictionary of mathematical terms used in English. – Washington : Mathematical Association of America, 1994. – 261 p.
34. Semantic Web Interest Group [Virtual Resource]. – Access Mode : URL : <http://www.w3.org/2001/sw/interest/>. – Title from Screen. – Date of Access : 9 October 2010.
35. Shelley Powers. Practical RDF. – Inc. : O'Reilly Media, 2003. – 331 p.
36. Structured Query Language (SQL) [Virtual Resource]. – Access Mode : URL : [http://databases.about.com/od/sql/Structured\\_Query\\_Language\\_SQL.htm](http://databases.about.com/od/sql/Structured_Query_Language_SQL.htm). – Title from Screen. – Date of Access : 7 October 2010.
37. System.Web.UI.WebControls Namespace [Virtual Resource]. – Access Mode : URL : <http://msdn.microsoft.com/en-us/library/8bhzw6t.aspx>. – Title from Screen. – Date of Access : 9 October 2010.
38. The Ontology Inference Layer OIL [Virtual Resource]. – Access Mode : URL : <http://xml.coverpages.org/OIL-inference.pdf>. – Title from Screen. – Date of Access : 7 October 2010.
39. The Semantic Web [Virtual Resource]. – Access Mode : URL : <http://semanticweb.org/>. – Title from Screen. – Date of Access : 24 September 2010.
40. The Syntax of CycL [Virtual Resource]. – Access Mode : URL : <http://www.cyc.com/cycdoc/ref/cycl-syntax.html>. – Title from Screen. – Date of Access : 24 September 2010.
41. The World Wide Web Consortium (W3C) is an international community that develops standards to ensure the long-term growth of the Web. [Virtual Resource]. – Access Mode : URL : <http://www.w3.org/>. – Title from Screen. – Date of Access : 24 September 2010.
42. Uniform Resource Identifiers (URI) Working Group [Virtual Resource]. – Access Mode : URL : <http://labs.apache.org/webarch/uri/> – Title from Screen. – Date of Access : 18 September 2010.
43. Uniform Resource Locators (URLs) [Virtual Resource]. – Access Mode : URL : <http://www.utoronto.ca/web/HTMLdocs/NewHTML/url.html> – Title from Screen. – Date of Access : 24 September 2010.
44. Uniform Resource Names (URN) Name space Definition Mechanisms [Virtual Resource]. – Access Mode : URL : <http://www.ietf.org/rfc/rfc3406.txt> – Title from Screen. – Date of Access : 7 October 2010.
45. Wache H. Ontology-Based Integration of Information: A Survey of Existing Approaches // Proceedings of the IJCAI-2001 Workshop: Ontologies and Information Sharing. – Seattle, WA, 2001. – P. 108–117.
46. Web Ontology Language: OWL [Virtual Resource]. – Access Mode : URL : <http://www.iro.umontreal.ca/~lapalme/ift6281/OWL/CostelloQuickIntroOwl.pdf> – Title from Screen. – Date of Access : 7 October 2010.
47. World Wide Web Search Portal [Virtual Resource]. – Access Mode : URL : <http://www.worldwideweb.com/> – Title from Screen. – Date of Access : 7 October 2010.