ДІРЕНКО І.С., ОЛЕЦЬКИЙ О.В.,
Національний університет "Києво-Могилянська академія"

СИСТЕМА УПРАВЛІННЯ ВМІСТОМ ВЕБ-РЕСУРСІВ НА ОСНОВІ ОНТОЛОПЧНО-ДОКУМЕНТНОГО МОДЕЛЮВАННЯ

Сьогодні стає все більш зрозуміло, що сучасні інформаційні системи, зокрема, веб-орієнтовані системи, повинні все більше спиратися на семантику предметної області, на знання про неї. Якщо основу знань системи складає онтологія, то і система, яка базується на знаннях, повинна бути онтологічно орієнтованою.

Водночас характерною особливістю веб-орієнтованих інформаційних систем, зокрема систем управління контентом веб-сайтів, є наявність великої кількості документальної інформації, причому самі ці документи часто генеруються динамічно, на ос-

нові тих чи інших процедур. Тому веб-орієнтована інформаційна система, яка базується на знаннях, повинна бути орієнтована на роботу як з онтологіями, так і з документами і множинами документів.

Тому в основі моделі інформаційної бази веб-орієнтованої системи повинні лежати дві вузлові компоненти: онтологія предметної області та множина документів, а також зв'язки між цими компонентами, іншими словами, повинен бути побудований граф, який складається не з логічно розрізнених документів, що характерно для більшості сучасних веб-орієнтованих систем, а з описів реальних класів предметної області, їх екземплярів та зв'язків між ними, а також пов'язаних з ними документів. Таким чином, можна говорити про проблему "занурення" множини документів, які можуть бути статичними або генеруватися динамічно, в загальну семантику предметної області [1].

Більш детальна формалізація зв'язків між семантикою предметної області та множиною документів може бути здійснена на основі формальної моделі онтології [2]. Слід також зазначити, що база знань, яка лежить в основі інтелектуальної інформаційної системи, може розглядатися як наповнення власне онтології як концептуальної схеми [3, 4].

Формально, ми розглядаємо модель інформаційного наповнення веб-орієнтованої системи як трійку $M=<W^*,D,L>$, де ^-онтологія предметної області, W^*- розширена онтологія, наповнення онтології W конкретними екземплярами класів (фактично - база знань), D-множина документів; /.-множина зв'язків між W^* та D.

На найбільш загальному рівні, в термінології UML [5] власне онтологія пов'язана з діаграмою класів, а наповнення бази знань - з діаграмою екземплярів. Але об'єктно-орієнтована реалізація онтологічного підходу, незважаючи на спорідненість цих напрямків, має свою специфіку. Зокрема, в [6] зазначається, що розробка онтологій значно відрізняється від проектування класів і зв'язків в об'єктно-орієнтованому програмуванні. Об'єктно-орієнтоване програмування зосереджується в основному на методах класів - програміст приймає проектні рішення, базуючись на операторних методах класу, тоді як розробник онтології приймає рішення, ґрунтуючись на структурних властивостях класу. В результаті структура класу і зв'язки між класами в онтології значно відрізняються від структури подібної предметної області в об'єктно-орієнтованій програмі.

В [1] показано, що значеннями функцій інтерпретації онтології Зможуть бути елементи множини документів D. При цьому мова повинна йти не тільки власне про онтологію, але і про всю базу знань, побудовану на її основі. Більш точно, якщо онтологія розглядається як трійка $<Q, Я, F>$, де Q - множина класів, які відповідають поняттям предметної області, $Я$ - множина зв'язків між ними, а F - множина функцій інтерпретації, то розширена онтологія описується як трійка $<Q^*, Я^*, F^*>$, де Q^* - множина класів разом з їх екземплярами, $Я^*$- множина зв'язків між цими елементами, а P - множина функцій інтерпретації, визначених у найпростішому випадку на елементах з Q^*, $Я^*$ та $Q^* \times Я^* \times Q^*$. Тоді елементи D можуть бути значеннями функцій з P. Іншими словами, будемо вважати документ d релевантним відносно W^*, якщо існують хоча б один вузол и/та функція інтерпретації f, такі що $d=f(w)$.

Важливими елементами веб-орієнтованої інформаційної системи повинні стати класи, які реалізують функції інтерпретації і можуть використовуватися для динаміч-

ного формування документів, а також класи, які забезпечують відображення отриманих документів.

Наведене співвідношення може стати основою для динамічного формування переліку споріднених документів. Для документа *d=f(w)* спорідненими документами можуть вважатися, зокрема, такі:

- всі документи *s* такі, що *s=h(w), he F ** (тобто документи, пов'язані з тим самим вузлом іншими функціями інтерпретації);
- всі документи *s* такі, що *s=g(u),* де функції інтерпретації *д* пов'язані з *f,* вузли *и* пов'язані з *w.*

Принципово важливим є те, що зв'язки і можливі переходи між документами повинні перш за все визначатися зв'язками між класами онтології та їх екземплярами. Тому одним з найбільш ключових компонентів стає система навігації*, яка на основі онтології забезпечує динамічне формування навігаційних графів [1], які визначають можливі переходи по веб-сайту.

Якщо ввести міри, які характеризують семантичну близькість понять онтології, а також ступені важливості функцій інтерпретації, це дозволить здійснити ранжування документів за мірою їх спорідненості до даного. Цікаво також дослідити, як введені таким чином міри спорідненості співвідносяться з традиційними підходами до визначення мір схожості документів на основі аналізу власне текстів [7, 8, 9 та ін.].

У цьому контексті стає очевидним, що проблема динамічного формування документів, споріднених до даного, найтіснішим чином пов'язана з проблемою підвищення повноти та релевантності інформаційного пошуку. Дійсно, незалежно від того, чи користувач вийшов на певний вузол онтології в процесі навігації по сайту, чи він ввів відповідне ключове слово в полі введення - мова йде про формування переліку документів, пов'язаних з даним вузлом, і ранжування цих документів за мірою релевантності.

Іншою важливою вимогою до веб-орієнтованих інформаційних систем є необхідність використання сучасних стандартів і технологій з огляду на логіку та перспективи їх розвитку. Ідеться перш за все про проект Semantic Web [10] орієнтований на обмін даними в середовищі WWW та на їх повторне використання і особливо - на опис семантики, змісту даних, які зберігаються на веб-ресурсах. Semantic Web спирається перш за все на наступні рекомендації, затверджені консорціумом W3C:

- розширена мова розмітки XML;
- структура опису ресурсу RDF [11] (Resource Description Framework). Ця структура забезпечує можливість для взаємодії застосувань на основі аналізу описів веб-документів. RDF Schema, в свою чергу, допомагає поєднувати ці описи в єдиний словник. Специфікація RDF надає потужну інфраструктуру для підтримки обміну знаннями в Інтернет.
- мова веб-онтологій OWL [12] (Ontology Web Language), що надає інструментарій для побудови структурованих веб-орієнтованих онтологій, які легко інтегруються та забезпечують можливість взаємодії даних між різними групами.

З огляду на описані вище принципи проектування веб-орієнтованих інформаційних систем, які повинні брати до уваги зв'язки між онтологією предметної області та множиною документів, а також з використанням наведених вище рекомендацій, роз-

роблено прототип системи управління вмістом веб-ресурсів, який забезпечує наступні можливості:

1. Система генерує навігаційні графи на основі онтологічного опису предметної області, може переходити від однієї навігаційної структури до іншої та від одного списку відповідного поняття предметної області до іншого в залежності від задания потрібної системи відношень та функцій інтерпретації. При цьому залишається можливість жорсткого задания системи навігаційних посилань на основі відношення *"рубрика/підрубрика"*.
2. Система працює з онтологією, яка включає три логічно виокремленні рівні:
   - рівень даних або онтологія предметної області: описує базові поняття, класи та зв'язки між ними, а також конкретні сутності ПрО.
   - рівень структури, що забезпечують навігацію користувача по сайту;
   - рівень представлення даних або онтологія структурних шаблонів, що керує відображенням інформації у вигляді, зручному для користувача.
3. Система спирається на XML, RDF та OWL
4. Базова онтологія розроблялася з використанням редактора онтологій «Рroiйдй» [13], крім того, система дозволяє адміністратору редагувати онтології за допомогою зручного веб-інтерфейсу.

Для того щоб описати модель системи управляння веб-сайтом, що базується на базі знань, необхідно спочатку описати вимоги що накладаються на онтологію. Онтологічна база знань разом з редактором онтологій має відповідати наступному переліку вимог, сформульованому на основі [14]:

1. *Структура:* онтологія має відображати всю структуру інформаційного ресурсу.
2. *Вміст:* онтологія має надавати можливість акумулювати різні типи інформації, а також дозволяти посилання на інші онтології.
3. *Імпорт та експорт:* вміст онтології має бути легко імпортованим або експортованим в систему і з системи відповідно.
4. *Життєвий цикл інформації:* онтологія має надавати можливості для точного представлення та ілюстрації всіх етапів життєвого циклу.
5. *Відображення:* онтологія має забезпечувати легкість внесення змін до представлення інформації на сайті. Для цього до розробленої на попередньому етапі онтології було додано клас Layout.

Структура сайту визначається екземплярами одного класу - класу сторінок (Pages). Таким чином, до неї можна звернутись як до дерева, кореневому елементу якого відповідає початкова сторінка. Завданням онтології є змоделювати ієрархічну деревовидну структуру і вказати для кожного елементу його рівень та позицію на цьому рівні. Для цього в класі сторінок вводяться додаткові параметри (властивості), такі як:

- *Батьківський елемент:* тобто кожен елемент структури, крім кореневого, має батьківський елемент, який визначає рівень вкладеності елементу в навігаційній структурі.
- *Позиція:* властивість, що відповідає за порядок відображення певного елементу на визначеному рівні.

- *Ресурс:* URI ресурсу (класу або індивіду онтології, що відображається в даному розділі структури).

Завдяки цим властивостями структура веб-сайту стає повністю керованою і підпорядкованою адміністратору. Більш того, додавання нових елементів до структури полягає лише в введенні нового елементу та визначенні для нього вищезгаданих параметрів.

Інформація про відображення на веб-сайті надається елементами класу Layout (Шаблон), який спеціально додається до онтології предметної області.

існують /два способи внесення контенту до веб-сайту. Перший спосіб - це внесення контенту автором через певний редактор, другий ~ імпортування зовнішньої онтології. У випадку імпортування, онтологія, що імпортується має бути відповідно адаптована, тобто доповнена класами Layout та Pages, що визначають зовнішнє відображення та структуру відповідно. Після того, як вказані класи буде визначено, залишається лише додати екземпляри та їх властивості, що й визначать структуру та зовнішній вигляд.

Маючи онтологію, що повністю відповідає вимогам до структури, можемо перейти до безпосередньої обробки OWL файлу. З цією метою було обрано pOWL [15] - бібліотеку для роботи з веб-онтологіями, що поширюється як продукт з відкритим кодом.

Враховуючи специфіку застосування, мовою програмування було обрано PHP, а оскільки бібліотека pOWL найбільше задовольняє всім функціональним вимогам і реалізована мовою PHP, то застосування саме цієї бібліотеки видавалось найбільш прийнятним.

Розроблена семантично орієнтована система, складається з трьох основних модулів:

1. Модуль pOWL бібліотеки: цей модуль надає системі функції для обробки та аналізу OWL дерева.
2. Модуль редактора онтологій: модуль що дозволяє користувачу, що не знається на структурі OWL редагувати онтологію.
3. Модуль системи управління навігацією: найбільш об'ємний модуль, що в свою чергу складається з декількох частин, зображених на рисунку 1.

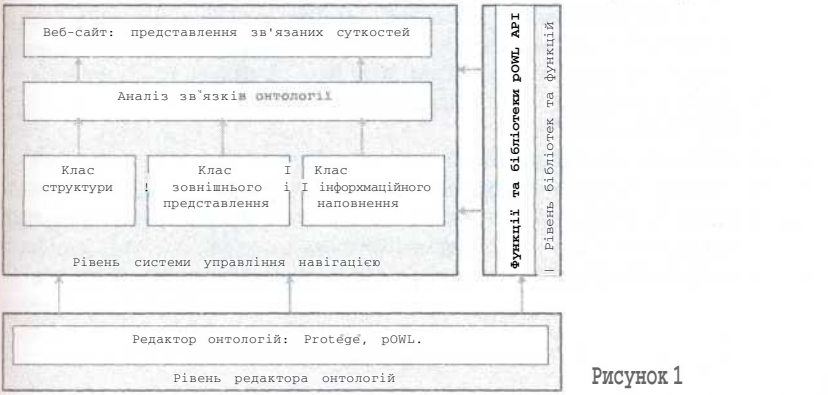

Рисунок 1

Таким чином, серед переваг семантично-орієнтованої системи над традиційними, крема в тому аспекті, який стосується навігації по веб-сайту, можна виділити наступні:

- мінімальна прив'язаність системи до конкретної предметної області. В основі інформаційного наповнення конкретного веб-ресурсу повинна лежати онтологія, яка аналізується системою, і на основі цього автоматично створюється прототип ресурсу;
- орієнтація на знання, які можуть змінюватись та доповнюватись з часом;
- можливість використання розробленої онтології іншими ресурсами;
- можливість динамічного формування на основі онтології предметної області навігаційних графів та інших елементів системи.

ЛІТЕРАТУРА

1. Олецький О.В. Застосування формальних моделей онтологій для формалізації інформаційних потоків у системах управління контентом. //Теоретичні та прикладні аспекти побудови програмних систем. Матеріали міжнародної конференції TAAPSD'2005, Київ, 7-9 грудня 2005 р. - С 26-29.
2. Гаврилова Т.А., Хорошевский В.Ф. Базы знаний интеллектуальных систем.- СПб:Питер, 2000. - 384 с
3. Андон Ф.И., Яшунин А.Е., Резниченко В.А. Логические модели интеллектуальных информационных систем. - К.:Наукова думка, 1999. - 398 с.
4. Проскудіна Г.Ю., Овдій О.М. Онтології в інформаційних системах. //Теоретичні та прикладні аспекти побудови програмних систем спеціальний випуск Вісник Київського національного університету ім.Т.Г.Шевченка, 2004. - С.164-169.
5. Буч Г., Якобсон А., Рамбо Дж. UML. ~ СПб.:Питер, 2006. - 736 с
6. Noy N., McGuinness D. Ontology Development 101: A Guide to Creating Your First Ontology// Stanford Knowledge Systems Laboratory Technical Report KSL-01-05 and Stanford Medical informatics Technical Report SMI-2001-0880, March 2001.
7. Сэлтон Дж. Автоматическая обработка, хранение и поиск информации. - М.Со-в.радио, 1973.-560 с.
8. Ландэ Д.В. Поиск знаний в Интернет. - М.:Изд. дом "Вильяме", 2005. - 272 с.
9. Гриценко В.И., Духновская К.К., Урсатьев А.А. Поисковый сервис. Проблемы, технологии, перспективы. //УСиМ, 2006, №2. - С. 81-92.
10. World Wide Web Consortium Specifications www.w3c.org
11. http://www.w3.org/RDF/
12. http://www.w3.org/2004/OWL/
13. Редактор онтологій ProШдй http://protege.stanford.edu/
14. Suren A. pOWL - A Web Based Platform for Collaborative Semantic Web Development // powi.sourceforge.net/swc/powl_usage.pdf

Semantic Web Development Platform - pOWL, www.powLsourceforge.net