О.В.Олецький, канд.техн.наук
Національний Університет "Києво-Могилянська Академія", ф-т інформатики
(Україна, 04070, Київ, вул.Сковороди,2,
тел. (044) 425-77-53, E-mail: oletsky@ukma.kiev.ua)

ДО ПРОБЛЕМИ КОМП'ЮТЕРНОГО МОДЕЛЮВАННЯ ТА АНАЛІЗУ НЕДОСТАТНЬО СТРУКТУРОВАНИХ ДАНИХ

Постановка задачі

Характерною особливістю сучасних розподілених систем є принципова гетерогенність джерел даних. Значна частина інформації є недостатньо структурованою, і доводиться мати справу не лише з реляційними базами даних, але й з документами та артефактами, які мають відношення до певних аспектів предметної області, а також з фрагментами даних, для яких неможливо чітко зафіксувати структуру та концептуальну схему.

Тому сьогодні на перший план виходить не стільки аналіз власне даних, але й аналіз метаданих (описів даних), а також моделей їх формування та можливих форм їх подання. На досить загальному рівні абстракції фрагменти даних, які присутні в інформаційній системі, слід розглядати як результат деякого відображення

$$q=h(S,D) \qquad (1)$$

де q – артефакт, що розглядається; D – власне дані; S – певний шаблон, який описує метадані та їх використання; h – функція, яка задає спосіб породження конкретного подання даних на основі їх поєднання з метаданими. Істотного значення набуває і вивчення ролі документів та артефактів у контексті можливих багатошарових архітектур інформаційних систем [1,2]. Ці питання сьогодні досліджені недостатньо, і дана робота спрямована на формалізацію моделей, які дозволяли б проводити аналіз відповідних проблем та розвивати підходи до їх вирішення.

В [3-5] розглянуто формалізації, які дозволяють описати відповідність між онтологією як моделлю предметної області [6,7] і множиною документів, та у більш загальній постановці задачі – артефактів системи. Більш точно, якщо онтологія розглядається як трійка $<Q, R, F>$, де Q – множина класів, які відповідають поняттям предметної області, R – множина зв'язків між ними, а F - множина функцій інтерпретації, то розширена онтологія описується як трійка $<Q^*, R^*, F^*>$, де Q^* - множина класів разом з їх екземплярами, R^*- множина зв'язків між цими елементами, а F^* – множина функцій інтерпретації, визначених у найпростішому випадку на елементах з Q^*, R^* та Q^* x R^* x F^*. Тоді елементи з множини документів D можуть бути значеннями функцій з F^*. Іншими словами, будемо вважати документ d релевантним відносно W^*, якщо існують хоча б один вузол w та функція інтерпретації f, такі що $d=f(w)$.

Таким чином, модель предметної області є, з одного боку, основою програмного проекту, а з іншого – може розглядатися як об'єктно-орієнтована

реалізація онтології – концептуальної моделі предметної області. Наведені вище співвідношення дозволяють інтегрувати до моделі предметної області артефакти, які пов'язуються з окремими вузлами онтології. Крім того, вони часто можуть формуватися динамічно на основі заданих операцій над окремими вузлами або підграфами розширеної онтологією. Таким чином, виникає задача побудови формалізованих моделей, на основі яких можна описувати процес динамічного формування документів.

У цьому контексті необхідно звернути увагу на XML [8] – загальну і дуже поширену рекомендацію щодо поєднання даних і метаданих, причому останні мають форму розмітки. Ця рекомендація орієнтована на чітке виділення окремих фрагментів даних і чітке відокремлення одного фрагмента від іншого. В контексті співвідношення (1) XML-запис – це деяке відображення певного змісту; можна казати про те, що конкретна форма цього відображення визначається певним проміжним рівнем, і моделі використання метаданих і утворюють цей рівень.

Але специфікації XML носять формальний характер і не дають відповіді на вищезгадані запитання. Як уже зазначалося, важливе значення мають моделі, які дозволяють отримувати не тільки власне дані, а й метадані. Питання про поєднання цих компонент є дуже неоднозначним, і це стосується навіть найпростішого і найбільш дослідженого випадку – реляційно-подібних фрагментів даних (тобто якщо можна встановити взаємно однозначну відповідність між окремими елементами XML-документа та кортежами деякого відношення реляційної бази даних). Це призводить до того, що останнім часом інтенсивно розвиваються форми подання даних, альтернативні до XML – наприклад, JSON.

Важливою видається ще одна проблема. Різні форми подання документів, породжені різними моделями, навіть якщо між ними існує взаємно однозначне відображення, є дуже різними за своїми можливостями. Можна навести багато прикладів, коли при одній формі подання певні задачі розв'язувати легко, а при іншій – набагато складніше, а інколи і просто неможливо.

Розглянемо найпростіший приклад – реляційно-подібний фрагмент даних. Йдеться про відображення інформації про певну кількість екземплярів деякого класу, причому всі вони мають просту однотипну структуру. Змістовно такий набір даних можна описати як матрицю даних $M=\{M_{ij}\}$, де M_{ij} – значення j-ї ознаки для i-го об'єкта. Крім такої матриці, потрібно зберігати інформацію про метадані, перш за все – про назви ознак, у вигляді вектора $V=\{v_j\}$, v_j – назва j-ї ознаки. Нехай, далі, K – назва класу, екземплярами якого є ці об'єкти. При цьому очевидним стає те, що компоненти вектора V можуть бути отримані на основі безпосереднього аналізу онтології предметної області.

На основі "класичної" і найбільш вживаної моделі формування XML-подання така інформація матиме вигляд:

<list>
<K>
<v_1> M_{11} </v_1>
...
<v_n> M_{1n} </v_n>
</K>

...
<K>
<v_1> M_{q1} </v_1>
...
<v_n> M_{qn} </v_n>
</K>

</list>,

тобто кожний елемент матриці M_{ij} супроводжується дескриптором, а самі ці дескриптори повторюються. Очевидною є велика надлишковість такої форми зберігання. Більш економним було б зберігати матрицю даних *M* окремо від вектора *V*. Таке подання могло б, наприклад, мати вигляд

<K>
<description> *вектор ознак* </description>
<data> *матриця даних*</data>
</K>

Друге подання, очевидно, є значно більш економним, ніж перше. Крім того, для нього можна застосовувати інші ефективні алгоритми стиску даних. Ефективним може виявитися, зокрема, використання методик інтегрального перетворення Карунена-Лоева [9, 10, 11], яке забезпечує оптимальний стиск даних за критерієм мінімізації середньоквадратичної похибки.

З іншого боку, перша (“класична”) форма подання, як правило, виявляється більш зручною для аналізу даних іншими програмами, а також людьми. Для неї суттєво полегшуються і інші операції, зокрема створення XML-схем та перевірка на їх основі дійсності документів.

Таким чином, для ефективного використання XML та подібних форматів для роботи з недостатньо структурованими даними видається доцільним застосовувати методи, пов’язані з побудовою та аналізом деякого графа – графа можливих трансформацій, вузлами якого є різні форми подання інформації та (або) моделі їх породження, а дуги – операції переходів. Для підтримки таких переходів природно використовувати інструментальні засоби, пов’язані з XSLT або XQuery. Крім того, з кожним вузлом пов’язується набір операцій, які зручно виконувати при даній моделі подання. Тоді розв’язання задач можна описати як деякий процес пошуку або поширення активації на графі, і загальну схемуа побудови і аналізу XML-документів на основі цієї методики можна охарактеризувати наступним чином.

1. Будується граф можливих трансформацій.
2. Задається критерій зупинки процесу пошуку.
3. Здійснюється процес пошуку, який продовжується до досягнення критерію зупинки.

На даний момент розроблений програмний прототип, який дозволяє продемонструвати деякі з описаних можливостей.

1. *Фаулер М.* Архитектура корпоративных программных приложений. – М.:Издательский дом «Вильямс», 2004. – 544 с.
2. *Нильссон Дж.* Применение DDD и шаблонов проектирования: проблемно-ориентированное проектирование приложений с примерами на C# и .NET. - М.:Издательский дом «Вильямс», 2008. – 560 с.
3. *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.
4. *Діренко І.С., Олецький О.В.* Система управління вмістом веб-ресурсів на основі онтологічно-документного моделювання //Теоретичні та прикладні аспекти побудови програмних систем. Матеріали міжнародної конференції TAAPSD'2006. Київ, грудень 2006 р. – С.171-176.
5. *Олецький О.В.* До проблеми онтологічно-орієнтованого пошуку в інформаційних системах. // Теоретичні та прикладні аспекти побудови програмних систем. Матеріали міжнародної конференції TAAPSD'2007. Бердянськ, 4-9 вересня 2007 р. – С.73-77.
6. *Гаврилова Т.А., Хорошевский В.Ф.* Базы знаний интеллектуальных систем. – СПб: Питер, 2000. – 384 с.
7. *Плескач В.Л., Рогушина Ю.В.* Агентні технології. - К.: Київ. нац. торг.-екон. ун-т, 2005. – 338 с.
8. *http://www.w3.org/TR.*
9. *Фукунага К.* Введение в статистическую теорию распознавания образов. - М.:Наука, 1979. - 368 с.
10. *Олецкий А.В.* О применении интегрального разложения Карунена - Лоэва при моделировании динамических систем. // УСиМ, 1999, №2. - С.12-15.
11. *Олецкий А.В.* Основные свойства и практические применения базовой квадратурной схемы интегрального разложения Карунена-Лоэва. // Моделювання та інформаційні технології. Вип.27. - Київ, 2004. - С. 113-120.