

Отже, СЗІ БД складається із:

- об'єктів та засобів СУБД (рівень СУБД);
- об'єктів та засобів БД (рівень схеми БД із засобами СЗІ);
- правил роботи із СЗІ та БД.

Необхідними вимогами для побудови безпечного середовища збереження та опрацювання даних є продумане проектування ІС, схеми БД, методик опрацювання інформації та функціональних обов'язків, попередня класифікація та оцінка даних, нормативна документація.

Напрямок подальших досліджень є побудова формальної моделі СЗІ бази даних, з урахуванням імовірностей причин втрати інформації та моделей можливих порушень безпеки інформації у БД та реалізація формальної моделі СЗІ у вигляді програмних моделей для захисту промислових СУБД.

1. Ильницкий А.Е., Шорошев В.В., Обзор зарубежных аппаратно-програмных средств защиты компьютерных систем // Бизнес и безопасность №6. 1998. 2. Тарасов Д.О., Забезпечення цілісності даних у реляційних структурах // Вісник ДУ "Львівська політехніка" 1999 №383. С. 213-226. 3. Дейт К. Дж. Введение в системы баз данных. :Пер. с англ. 6-е изд. К., 1998. 4. X. Qian and T.F. Lunt. Tuple-level vs. element-level classification. Database security, VI: Status and Prospects, North-Holland, 1993. Pp. 301-315.

УДК 681.3.06

**М.Ю.Щербина**

*Військовий інститут при НУ "Львівська політехніка"*

## **ДЕЯКІ МЕТОДОЛОГІЇ МОДЕЛЮВАННЯ ІНФОРМАЦІЙНИХ СИСТЕМ ДЛЯ INTERNET/INTRANET**

© М.Ю.Щербина, 2000

**Contemporary approaches for modeling Web information systems (WIS) are considered. The RMM and Extended RMM methodologies based on entity-relationship model are described in detail. Some enhancements of relational methodologies are proposed.**

Методології моделювання систем гіпермедіа почали активно розробляти на початку 90-х років, особливо після широкого впровадження мови HTML та інших Web-технологій. Методології базуються на різних фундаментальних засадах, наприклад, на моделі "сутність-зв'язок" (*entity-relationship model*) [1, 2] або на принципах об'єктно-орієнтованого моделювання. На принципах ER-моделювання основані, зокрема,

*Relational Management Methodology (RMM)* [3] та *Extended RMM* [4, 5], а на об'єктній орієнтації *Object-Oriented Hypermedia Design Method (OOHDM)* [6].

Використання сучасних Web-технологій (CGI, Java, DHTML, інтеграція з базами даних) дозволяє створювати складні *інформаційні системи для Web (WIS)*. Великою перевагою WIS є зручність у користуванні (оскільки як клієнтське ПЗ використовуються звичайні Web-браузери, такі як Microsoft Internet Explorer або Netscape Navigator). Більшість користувачів вже знайомі з принципами навігації по World Wide Web, тому зможуть швидко засвоїти принципи роботи з будь-якою WIS. Використання стандартних браузерів надає WIS всі переваги системи з “тонким” клієнтом і робить інформаційну систему практично платформно-незалежною. Всі ці чинники роблять Web ідеальною платформою при створенні інформаційних систем для Intranet-мереж. Для розв'язання задач віддаленого доступу (через Internet) до інформаційних ресурсів WIS взагалі є поза конкуренцією.

Останнім часом Web-технології стали стандартом де-факто при створенні систем гіпермедіа. Мова HTML на початку використовувалась лише для подання гіпертекстової інформації, проте нові стандартні формати представлення аудіовідеоінформації, разом із застосуванням Java та технології Dynamic HTML, дають змогу ефективно використовувати можливості мультимедіа.

Сучасні засоби розробки Web-сайтів використовуються лише на останніх етапах створення WIS — для створення графіки, генерації HTML-сторінок, аналізу гіперпосилань тощо. На жаль, практично не існує CASE-засобів, які враховують власне специфіку Web-систем. Тому з'явилися методології, що дозволяють систематизувати процес створення WIS, в тому числі й для початкових рівнів абстракції.

Розробка WIS відповідно до OOHDM складається з чотирьох базових стадій: концептуальний дизайн, навігаційний дизайн, створення абстрактного інтерфейсу та реалізація [1]. Вони виконуються в змішаному інкрементному, ітеративному та заснованому на прототипах стилях. На кожній стадії набір об'єктно-орієнтованих моделей, які описують відповідні аспекти дизайну, будуються чи поширюються на основі попередніх ітерацій. На цьому завершимо розгляд OOHDM, оскільки предметом даної статті є реляційні методології.

Ключовими для реляційних методологій створення WIS є поняття *сутностей (entity)* та *зв'язків (relationship)*. В даний час модель сутність-зв'язок (ER-модель) широко використовується при розробці реляційних баз даних. Базовими з даної тематики є праці Чена, зокрема [2]. ER-модель дозволяє ефективно зобразити реальні об'єкти довільної предметної галузі та зв'язки між ними. Тому її можна успішно використати не лише для розробки баз даних, а й при створенні систем гіпермедіа (в тому числі WIS).

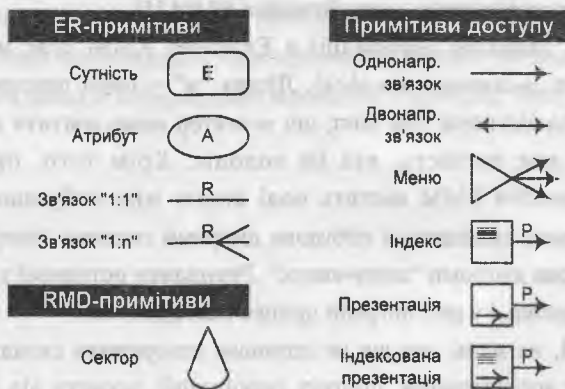
Зауважимо, що в даній статті термін “*реляційна*” використовується для методологій, які засновані на ER-моделюванні, тобто він походить від слова “*relationship*” (зв'язок). Водночас термін “*реляційна база даних*”, який широко використовується в спеціальній літературі, походить від “*relation*” (відношення). Проте ця невідповідність у цьому

випадку не є критичною, оскільки в кінці статті автором пропонується методика *ARAMIS*, що використовує як ER-підхід, так і відношення в стандартному розумінні.

Базовою ідеєю реляційних (ER-підхід) методологій є той факт, що ER-діаграма конкретного проекту може автоматично використовуватись при розробці навігаційної структури WIS. Для простих випадків сутності завжди відповідатиме Web-сторінка, а зв'язку — гіперпосилання. Цей факт впливає з самої концепції ER-моделювання і легко пояснюється з психологічного погляду. Винятком будуть сутності з великим набором атрибутів, яким може відповідати декілька Web-сторінок. Крім того, деякі зв'язки можуть виявитись зайвими для навігації.

Методологія RMM побудована власне на базі щойно викладених принципів [3]. Основними стадіями розробки WIS з використанням RMM будуть *ER-проекування* (*ER Design*), *секторний дизайн* (*Slice Design*), *навігаційний дизайн* (*Navigational Design*), а також *розробка та реалізація інтерфейсу користувача* (*User-Interface Design and Construction*).

Термін “*slice*”, який будемо перекладати як “*сектор*”, використовується в RMM власне для розв'язання проблеми представлення сутностей з великим набором атрибутів. Сектор репрезентує деяку групу атрибутів даної сутності, які будуть зображатись на заданій Web-сторінці. Наприклад, для сутності “працівник” зручно побудувати два сектори: *БазоваІнфо*(*прізвище, ім'я, посада*) та *ПовнаІнфо*(*прізвище, ім'я, фото, біографія*). Навігація між секторами забезпечується за допомогою одно- та двонапрямлених зв'язків. Ці зв'язки називають *структурними*, на відміну від *асоціативних* зв'язків, які присутні в ER-діаграмі. Після побудови секторних діаграм вони підставляються в ER-діаграму системи, що проектується. Отримана діаграма носить назву ER<sup>+</sup>.



Основні примітиви RMM

Для навігації між різними сутностями в RMM використовуються три базові конструкції: *індекси* (*indices*), *презентації* (*guided tours*) та *меню* (*groupings*). Індекс відтворює перелік екземплярів даної сутності, до кожного з яких надається доступ. Презентація дає можливість пересуватись між блоками інформації за допомогою

посилань на попередній та наступний екземпляри сутності. Наприкінці презентація може перекинути користувача як на головну сторінку, так і на перший елемент списку (циклічна презентація). Меню використовують для гіперпосилань на сторінки, логічно не пов'язані з рештою сторінок системи — наприклад, на інші Web-сайти. На рисунку зображені основні графічні конструкції RMM.

Для зазначення екземплярів сутності, які фігуруватимуть в індексі чи презентації, використовують логічні предикати. Наприклад, предикат *Працівник(посада = "інженер")* вказує, що в індексі чи презентації з усіх працівників будуть представлені лише інженери.

На етапі розробки інтерфейсу користувача для кожного об'єкта з діаграми, яка була отримана на попередніх стадіях, вказується розташування на екрані та стиль. Більшість сучасних засобів розробки Web-сайтів можуть бути задіяні власне на цій (іноді також на попередній) стадіях. Отже, RMM дозволяє систематизувати процес створення WIS на всіх етапах розробки. Великою перевагою RMM, на відміну від багатьох інших методологій, є графічне представлення проекту. Діаграма проекту, завдяки своїй читабельності, дозволяє розробнику легко узгоджувати проект з замовником системи. Легкість інтеграції з реляційними базами даних природно впливає з моделі "сутність-зв'язок", яка покладена в основу RMM. При цьому остаточна генерація HTML може здійснюватись як статично, так і динамічно, що абсолютно не впливає на процес розробки системи.

RMM, звичайно, має й ряд недоліків. Найбільш серйозним є неможливість моделювання складних Web-сторінок — інформацію з різних сутностей не можна комбінувати на одному екрані. Недоліком також є принцип розробки "зверху-вниз". Повторне використання компонент стає неможливим на нижньому рівні процесу розробки. Для подолання згаданих недоліків до RMM були внесені певні удосконалення [4]. Отримана методологія дістала назву Extended RMM [5].

Репрезентація складної інформації в Extended RMM стає можливою завдяки введенню так званих *м-секторів (m-slice)*. Літера "м" у назві походить від російського слова "*матрешка*" та відзначає той факт, що м-сектор може містити вкладені м-сектори. М-сектор завжди має сутність, яка їм володіє. Крім того, процес розробки з використанням Extended RMM містить нові етапи: між побудовою ER-діаграми та інтерфейсу користувача знаходяться побудова діаграми системи "зверху-вниз", розробка м-секторів та побудова діаграми "знизу-вверх". Результати останньої з нових трьох стадій верифікуються з першою, і у разі потреби процес повторюється.

Extended RMM, на жаль, все ще не дозволяє створювати складні WIS, які містять засоби інтеракції з користувачем, пошуку інформації, захисту від несанкціонованого доступу тощо. Подібні задачі здебільшого розв'язують за допомогою CGI. Для розв'язання цієї проблеми автором розроблена нова методологія *Advanced Relational Approach to Modeling Internet/Intranet Information Systems (ARAMIS)*. Ця методологія створювалась незалежно від RMM на базі принципів, які були викладені в [7]. Разом з тим ARAMIS та RMM мають багато спільних рис. ARAMIS більшою мірою використовує реляційну алгебру, наприклад, сектор у RMM — це ні що інше, як деяка проекція

відношення (сутності). Аналогом м-сектора буде група пов'язаних відношень. У ARAMIS введені спеціальні типи відношень, які є концептуальною основою системи. Наприклад, HTML-формі, а, отже, і генерованому CGI-запиту, відповідатиме спеціальне відношення, атрибутами якого будуть елементи введення інформації, присутні у формі. В найпростішому випадку відношення складатиметься з одного кортежу, проте при використанні, наприклад, елементів <SELECT MULTIPLE> відношення міститиме вже декілька кортежів. Інформація, що стосується сесії окремого користувача WIS, також репрезентується спеціальним типом відношень, які фізично зберігаються за допомогою механізму HTTP Cookie. Такі властивості цієї методології дозволяють ефективно створювати достатньо складні інтерактивні WIS. У даний час методика ARAMIS удосконалюється для пристосування для ширшого кола задач.

1. Chen P. *Database Design Based on Entity and Relationship* // In S. Bing Yao, Editor, *Principles of Database Design, Volume 1: Logical Organizations, chapter 5.* Prentice Hall, Inc., Englewood Cliffs, New Jersey 07632, USA, 1985. P. 174-210. 2. Чен П. Модель "суцність-св'язь" – шаг к единому представлению данных // *Database Management Systems.* 3 (1995) (<http://www.case.ru>). 3. Isakowitz T., Stohr E., Balasubramanian P. *RMM: A Methodology for the Design of Structured Hypermedia Applications* // *Communications of the ACM.* 38, 8 (1995). P. 34-41. 4. Isakowitz T., Kamis A., Koufaris M. *Extending the Capabilities of RMM: Russian Dolls and Hypertext* // *Proceedings of the 30th Annual Hawaii International Conference on System Sciences, 1997.* (<http://rmm-java.stern.nyu.edu/rmm>). 5. Isakowitz T., Kamis A., Koufaris M. *The Extended RMM Methodology for Web Publishing.* (<http://rmm-java.stern.nyu.edu/rmm>). 6. Schwabe D., Rossi G., Barbosa S. *Structured Web Site Design.* (<http://www-di.inf.puc-rio.br/~schwabe/HT96/WWW-Workshop.html>). 7. Щербина М.Ю. Вимоги до CASE-систем проектування застосувань для Internet/Intranet, що взаємодіють із реляційними базами даних // *Інформаційні технології і системи, 1998, №1/2.* С. 203-208.