УДК 004.652.8, 004.94

**Рязанцев О.І., Барбарук В.М., Татарченко Г.О.**

# ПРОЕКТУВАННЯ СХОВИЩ ДАНИХ ЗА ДОПОМОГОЮ UML

*У сценаріях сховищ даних (СД, англ. Data Warehouse, DW) процеси вилучення, перетворення, завантаження даних (англ. Extraction, Transformation, Loading, ETL) відповідають за витяг даних з неоднорідних джерел операційних даних, їх перетворення (перетворення, очищення, нормалізацію тощо) та їх завантаження у сховище даних. У статті наведено формалізовану структуру для моделювання взаємозв'язків між джерелами даних та цілями їх видобутку в різних рівнях деталізації атрибутів. В якості інструмента моделювання застосовано уніфіковану мову моделювання (англ. Unified Modeling Language, UML), що дозволяє масштабувати дизайн сценарію у напряму як збільшення рівня деталізації атрибутів, так і їх зменшення.*
***Ключові слова:*** *карта даних, ETL, сховище даних, UML*

**Актуальність дослідження.** Процеси ETL у сценаріях СД відповідають за витяг даних з неоднорідних джерел операційних даних, подальшого їх перетворення (перетворення, очищення, нормалізацію тощо) та їх завантаження у сховище даних. Зазвичай СД містять дані з різних та неоднорідних джерел операційних даних, таких як реляційні бази даних, різноманітні файли, файли Інтернет (XML, веб-журнали), мультимедійні файли, тощо. Розробка та підтримка ETL- процесів є ключовим фактором успіху в проектах СД. І причин тому декілька. Найбільш відомими з них є їх критична маса часу - ; бо розробка ETL може зайняти до 80% часу розробки проекту усього СД [1,2]. Незважаючи на важливість розробки відображення джерел даних на структуру СД, на жаль, існує лише декілька моделей, котрі розробники можуть використовувати для цих цілей. Тим не менше [3, 4], донині немає моделі, яка може поєднувати бажаний рівень деталізації інтеграції даних моделювання на рівні атрибутів та широко прийнятий формалізм моделювання, хоча б такі, якими володіють ER (Entity Relations) моделі або UML. Однією з причин цього є те, що обидва ці формалізм просто не призначені для цього завдання. І навпаки, вони розглядають атрибути як другорядні, слабкі сутності, що мають описову роль головних об’єктів даних.

**1. Вступ.** Можна стверджувати, що сучасний спосіб моделювання СД є достатнім, і немає необхідності його розширювати, щоб зафіксувати відображення та перетворення на рівні атрибутів. І тому є певні причини:

- Елементи проектування СД мають аналогічний вплив на наступні етапи проекту СД як і креслення деталі для її подальшого виробництва. Одна з постійних задач проектувальників СД полягає у складанні документації кожного етапу проектуванні. Оскільки елементи проектування СД на початковому рівні є засобом комунікації ідей розробників, то задля їхній формалізмі слід використовувати широко розповсюдженні засоби (наприклад, UML або ER).

- Дизайн повинен відображати архітектуру СД таким чином, щоб мати можливість подальшого аналізу будь-яких наступних модифікацій СД. Виділення атрибутів та зв'язків між ними є елементом моделювання першого класу (англ. First-Class Modeling Elements, FCME). У той же час засоби, які зараз застосовуються на FCME використовують документацію, оформлену за допомогою нотаток UML.

Основна ідея дослідження полягає у застосовуванні підходу, який дозволяє відстежувати специфіку обробки ETL на різних рівнях деталізації за допомогою поширеного формалізму UML. Це реалізується додатковим представленням СД, яке називається діаграмою мапірования (англ. data mapping diagram). У цій новій діаграмі ми розглядаємо атрибути як FCME. Це дає нам можливість отримати уявлення про визначені моделі СД на різних рівнях деталізації. Звичайно, оскільки UML у класичному виконанні не підтримує новий тип діаграм, то ми, як і інші дослідники [5-9], вирішуємо цю проблему завдяки розробці та застосуванню додаткових модулів (англ. plug-in modules).

**2. Атрибути FCME в UML.** Як в моделях ER так і в UML, атрибути входять до складу «найменших» елементів (об'єкт в ER або клас у UML). Обмеженням є те, що неможливо створити зв'язок між двома атрибутами одного елементу (наприклад, при інтеграції даних, при обмеженнях щодо атрибутів, тощо). Вирішення ситуації полягає у відношенні до атрибутів ні як властивості елементу, а як самостійного елементу СД, тобто FCME. Ми зробили вибір на користь UML замість ER за умови її кращої адаптації до використання додаткових діаграм для проектування складної інформаційної системи.

Розкриємо зміст поняття FCME. Концептуально, FCME називають фундаментальні засади моделювання, на основі яких будуються складні моделі. Технічно, FCME включають власну ідентичність, і, можливо, обмеженість цілісності. У діаграмі класів UML два види елементів моделювання розглядаються як FCME. Класи, як абстрактні уявлення реальних сутностей, природно, знаходяться в центрі моделювання. Виступаючі в якості FCME, класи є автономними об'єктами, які також виступають як атрибути. Відносини між класами сприймаються асоціаціями. Асоціації також можуть бути організовані на рівні FCME. Незважаючи на те, що графічно клас асоціації відображається як сукупність асоціації та класу, це дійсно лише один елемент моделі [10]. Клас асоціації може містити свої атрибути або бути пов'язаним з іншими класами. І це не можливо зробити з атрибутами у класичному їхньому трактуванні.

Звичайно, щоб мати можливість атрибутам відігравати ту ж саму роль що й класу, ми пропонуємо представлення атрибутів як FCME в UML. У нашому підході класи та атрибути визначаються так же само, як і в класичному UML. Однак у тих випадках, коли необхідно розглядати атрибути як FCME, класи імпортуються на діаграму атрибуту/класу, де атрибути автоматично відображаються як класи; таким чином, користувач повинен лише визначити класи та атрибути один раз. У процесі імпорту з діаграми класів до діаграми атрибутів/класів ми звертаємось до класу, який містить атрибути як клас контейнера, і до класу, який представляє атрибут як клас атрибута.

**3. Діаграма відображення даних.** Після того, як ми запровадили механізм розширення, який дозволить UML обробляти атрибути як FCME, ми можемо переходити до розроблення засобів його використання. Представимо діаграму мапірования даних, яка є новою схемою, що спеціально налаштована для відстеження потоку даних у різних ступенях деталізації в середовищі СД. Діаграми відображення даних є додатковими до типових класів та діаграм взаємодії UML, і зосереджені на особливостях потоку даних та взаємозв'язках із задіяними сховищами даних. Особливою характеристикою діаграм відображення даних є те, що певний сценарій СД практично описується набором додаткових діаграм відображення даних, кожен з яких визначається різним рівнем деталізації. Представимо принциповий підхід до застосування таких додаткових схем відображення даних.

Для встановлення взаємозв'язків між елементами проектування СД з точки зору даних ми використовуємо поняття відображення. Загалом, коли два елементи проектування (наприклад, дві таблиці або два атрибути) мають однакову частину інформації, то з’являється можливість через певне перетворення чи фільтрацію встановити семантичний зв'язок між ними. У контексті СД ці відносини включають три логічні сторони: (а) суб'єкт постачальника даних (схема, таблиця або атрибут), який відповідає за створення даних для подальшого поширення, (б) споживач, який отримує дані від провайдера і (в) їх проміжне узгодження, яке передбачає спосіб здійснення відображення, а також будь-яке перетворення та фільтрацію.

Оскільки діаграма відображення даних може бути дуже складною, наш підхід передбачає можливість організувати її різними рівнями завдяки використанню пакетів UML. Наша багатопрофільна пропозиція складається з чотирьох рівнів (див. рис. 1)

На самому лівий частині рисунку 1 показана проста взаємодія між концептуальною схемою сховища даних (англ. Data Warehouse Conceptual Schema, DWCS) та концептуальною схемою джерела даних (англ. Source Conceptual Schema , SCS): це зафіксовано одним пакетом Mapping Data, і ці три елементи проектування складають діаграму відображення даних на рівні бази даних (так званому, Рівні 0). Якщо припустити, що у СД є три окремі таблиці, які ми хочемо заповнити, цей конкретний пакет картки даних репрезентує той факт, що існує три основних сценарії для наповнення СД, одна для кожної з цих таблиць. На рівні потоку даних (так званому Рівні 1) системи відносини даних між джерелами та цілями в контексті кожного з сценаріїв практично моделюються відповідним пакетом. Якщо ми збільшимо один з цих сценаріїв, наприклад, Mapping 1, ми можемо спостерігати його особливості з точки зору перетворення та очищення даних: дані джерела 1 перетворюються в два етапи (тобто вони пройшли два різних перетворення), як показано на рис.3. Слід зазначити, що використовується сховище проміжних даних, щоб утримувати вихід першого перетворення (Крок 1), перш ніж передати на другий (Крок 2). Нарешті, у правій нижній частині на рисунку 3 зображено спосіб яким атрибути пов’язані один з одним для зберігання даних, джерела 1 та проміжних даних. Звернемо увагу на те, що у випадку, коли ми моделюємо складне та величезне СД, атрибут перетворення моделюється на Рівні 3, тобто занурюється на один рівень униз для уникнення «забруднення» діаграми даних.

Конструкції, які ми використовуємо для діаграми зіставлення даних на різних рівнях, є такими:

- Бази даних та діаграми потоку даних (Рівні 0 та 1) використовують традиційні UML-структури. Зокрема, на цих діаграмах ми використовуємо (а) пакети для моделювання відносин даних та (б) прості залежності між включеними сутностями. Залежності вказують на те, що пакети відображення залежать від змін застосованих сховищ даних.

- Діаграма рівня таблиці (рівень 2) розширює UML з трьома стереотипами: (а) "Мапірування", що використовується як пакет, який містить в собі взаємовідносини даних між сховищами даних; б) "Вхід" і "Вихід", які пояснюють ролі провайдерів та споживачів для "мапірування".

- Діаграма на рівні атрибуту (рівень 3) також використовує кілька нових введених стереотипів, а саме: "Карта", "Обсяг карти", "Домен", "Діапазон", "Вхід", "Вихід" та "Середній рівень" для визначення відображення даних.

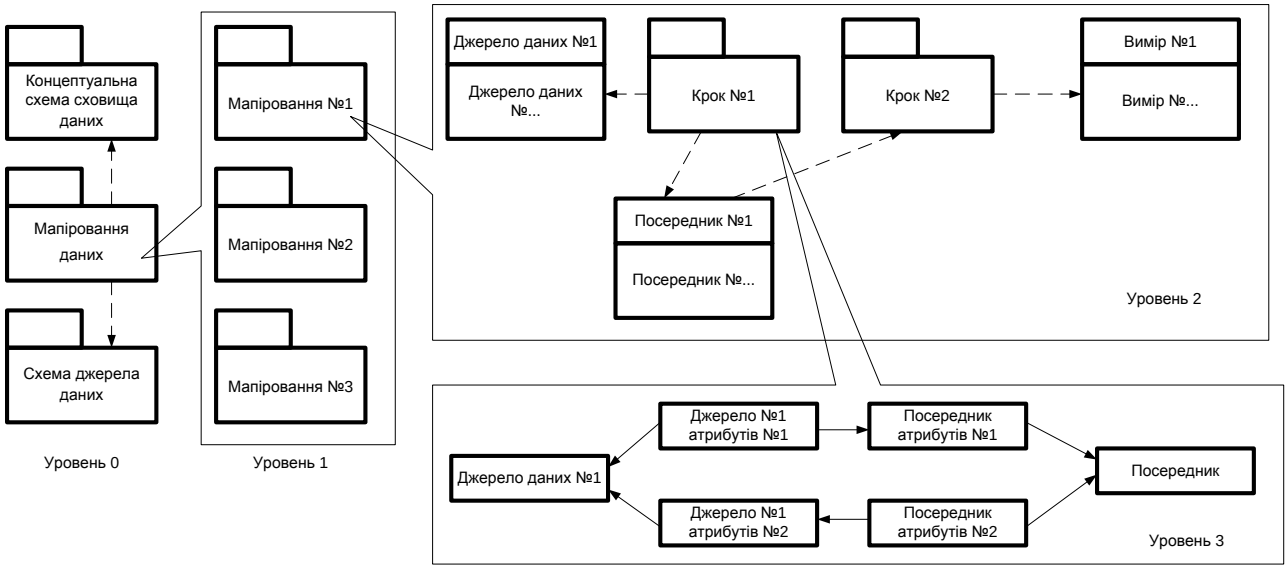


Рисунок 1. Рівні відображення даних

**4. Діаграма відображення даних на рівні атрибута.** Як вже було сказано, на рівні атрибута діаграма відображення даних включає в себе відносини між атрибутами класів, що беруть участь у відображення даних. На цьому рівні ми розробляємо два варіанти проектування:

- Компактний варіант: зв'язок між атрибутами представлений як асоціація, а семантична відповідність описується в нотаціях UML, доданих до атрибута цілі відображення.

- Формальний варіант: зв'язок між атрибутами представлений за допомогою об'єкта зіставлення, а семантична відповідність описується в визначенні тегу об'єкта відображення.

У першому варіанти формування діаграми відображення даних вона менш «забруднена», при цьому застосовується менше елементів моделювання, а семантика відображення даних виражається як UML-нотатки, які є простими коментарями, які не мають семантичного впливу. З іншого боку, розмір діаграми відображення даних, отриманих за другим варіантом, більший, з більшою кількістю моделюючих елементів та відносин, але семантика краще визначена як визначення тегів. На даному етапі дослідження ми зосередимось лише на компактному варіанті. У цьому варіанті взаємозв'язок між атрибутами представлений як асоціація, прикрашена стереотипом "Карта", а семантика відображення описується в нотатках UML, доданих до цільового атрибута відображення.

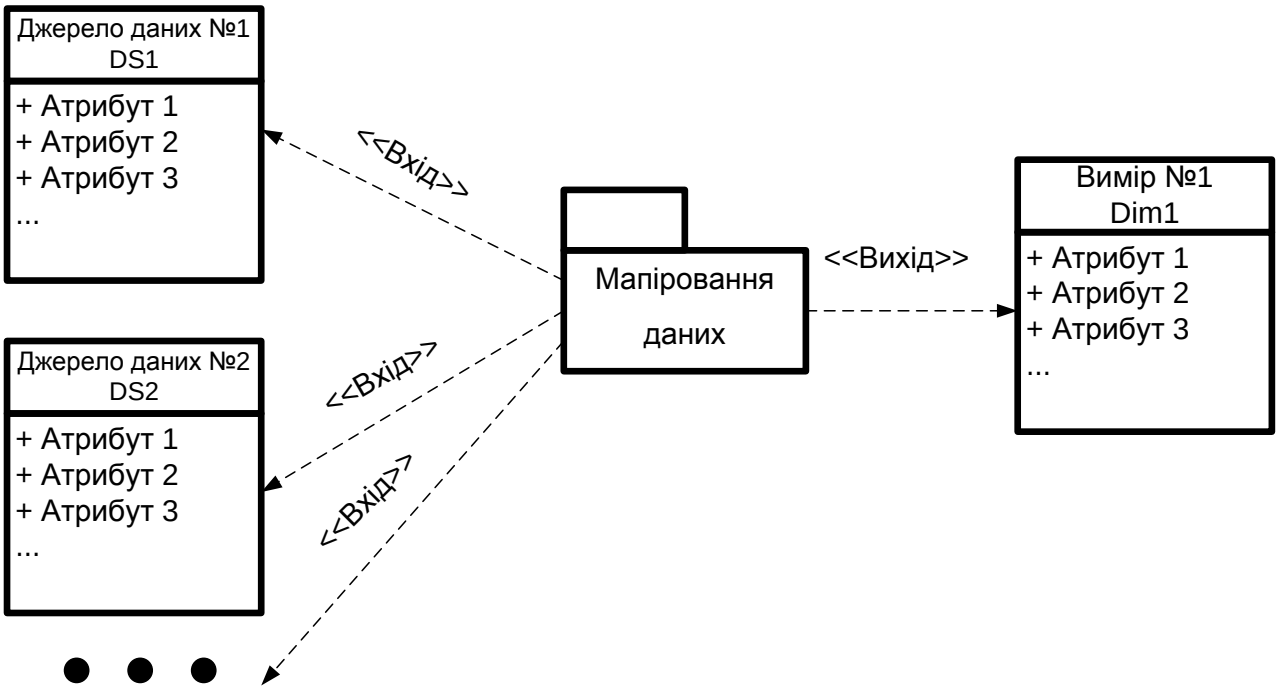


Рисунок 2. Рівень 2 діаграми відображення даних

Зміст схеми відображення пакунків з рис. 2 визначається таким чином:

– Класи DS1, DS2,. . . , і Dim1 імпортуються на діаграму відображення.

– Атрибути цих класів пригнічуються, оскільки вони відображаються як клас "Атрибут" у цьому пакеті.

– Класи "Атрибут" пов'язані за допомогою зв'язків асоціацій, і ми використовуємо властивість навігації для визначення потоку даних з джерел даних до СД.

– Асоціативні відносини доповнені стереотипом "Карта", щоб підкреслити значення цього співвідношення.

– UML-нотатки можуть бути доданими до кожного з цільових атрибутів, щоб зазначити те, як цільовий атрибут отримується з атрибутів джерела. Мова для формалізації нотатків - це вибір проектувальника (наприклад, підхід LAV або GAV [11]).

**Висновки.** У статті представлено основу для розробки зворотного етапу проектування СД (і відповідних процесів ETL) на підставі ключового спостереження, що це завдання, по суті, включає в себе вирішення специфічних задач при дуже низьких рівнях деталізації проекту СД. Зокрема, ми представили формалізовану основу для моделювання взаємозв'язків між джерелами та цілями в різних рівнях деталізації проекту (тобто від грубих відображень на рівні бази даних до детальних відображень між атрибутами на рівні атрибутів). На жаль, стандартні засоби моделювання, такі як ER-модель або UML, принципово не підтримують обробку суб'єктів низької деталізації (наприклад, атрибутів) як FCME. Тому, щоб формально досягти вищезгадану мету, ми розширили UML для моделювання атрибутів як FCME.

Хоча ми розробили представлення атрибутів як FCME в UML у контексті СД, ми вважаємо, що наше рішення може бути застосоване і в інших областях, наприклад, визначення індексів та матеріалізації представлень у базах даних, моделювання XML-документів, специфікації веб-служб та т.і.

**Література**


1. SQL Power Group: How do I ensure the success of my DW? Internet: http://www.sqlpower.ca/page/dw_best_practices (2002)

2. Strange, K.: ETL Was the Key to this Data Warehouse's Success. Technical Report CS-15-3143, Gartner (2002)

3. Vassiliadis, P., Simitsis, A., Skiadopoulos, S.: Conceptual Modeling for ETL Processes. In: Proc. of 5th Intl. Workshop on Data Warehousing and OLAP (DOLAP 2002), McLean, USA (2002) 14–21

4. Trujillo, J., Lujбn-Mora, S.: A UML Based Approach for Modeling ETL Processes in Data Warehouses. In: Proc. of the 22nd Intl. Conf. on Conceptual Modeling (ER'03). Volume 2813 of LNCS., Chicago, USA (2003) 307–320

5. Vassiliadis, P., Simitsis, A., Skiadopoulos, S.: Modeling ETL Activities as Graphs. In: Proc. of 4th Intl. Workshop on the Design and Management of Data Warehouses (DMDW'02), Toronto, Canada (2002) 52–61

6. Lujбn-Mora, S., Trujillo, J., Song, I.: Extending UML for Multidimensional Modeling. In: Proc. of the 5th Intl. Conf. on the Unified Modeling Language (UML'02). Volume 2460 of LNCS., Dresden, Germany (2002) 290–304

7. Lujбn-Mora, S., Trujillo, J., Song, I.: Multidimensional Modeling with UML Package Diagrams. In: Proc. of the 21st Intl. Conf. on Conceptual Modeling (ER'02). Volume 2503 of LNCS., Tampere, Finland (2002) 199–213

8. Lujбn-Mora, S., Trujillo, J.: A Comprehensive Method for Data Warehouse Design. In: Proc. of the 5th Intl. Workshop on Design and Management of Data Warehouses (DMDW'03), Berlin, Germany (2003) 1.1–1.14

9. Jarke, M., Lenzerini, M., Vassiliou, Y., Vassiliadis, P.: Fundamentals of Data Warehouses. 2 edn. Springer-Verlag (2003)

10. Object Management Group (OMG): Unified Modeling Language Specification 1.4. Internet: http://www.omg.org/cgi-bin/doc?formal/01-09-67 (2001)

11. Lenzerini, M.: Data Integration: A Theoretical Perspective. In: Proceedings of the Twenty-first ACM SIGACT-SIGMOD-SIGART Symposium on Principles of Database Systems, Madison, Wisconsin, USA (2002) 233–246



*In Data Warehouse (DW) scripts, extraction, transformation, loading, ETL processes are responsible for extracting data from heterogeneous sources of operational data, transforming them (transformation, purification, normalization). etc.) and their download to the data warehouse. The article presents a formalized structure for modeling the relationships between data sources and their extraction goals at different levels of attribute detail. As a simulation tool, the Unified Modeling Language (UML) has been applied, which allows scaling the design of the script in a direction, both as an increase in the level of detail of the attributes and their reduction.*


***Keywords:*** *data mapping, ETL, data warehouse, UML*



*В сценариях хранилищ данных (СД, англ. Data Warehouse, DW) процессы извлечения, преобразования, загрузки данных (англ. Extraction, Transformation, Loading, ETL) отвечают за извлечение данных из неоднородных источников операционных данных, их преобразования (преобразование, очистки, нормализацию и т.п.) и их загрузки в хранилище данных. В статье приведены формализованную структуру для моделирования взаимосвязей между источниками данных и целями их добычи на разных уровнях детализации атрибутов. В качестве инструмента моделирования используется унифицированный язык моделирования (англ. Unified*

*Modeling Language, UML), что позволяет масштабировать проектирование сценария в направлении как увеличение уровня детализации атрибутов, так и их уменьшения.*
***Ключевые слова:*** *карта данных, ETL, хранилище данных, UML*

**Рязанцев О. І. –** д.т.н., професор, проректор з науково – педагогічної роботи та міжнародної діяльності Східноукраїнського національного університету імені Володимира Даля, e-mail: mailto:ma_ryazantsev@snu.edu.ua

**Барбарук В.М.** – к.т.н., доцент, директор центру удосконалення освіти Східноукраїнського національного університету імені Володимира Даля, e-mail: barbaruk.viktor@gmail.com

**Татарченко Г.О.** - д.т.н., професор, завідувач кафедри міського будівництва та господарства Східноукраїнського національного університету імені Володимира Даля, email: tatarchenkogalina@gmail.com