

УДК 004.415.26:004.415.2

В. І. Гайдаржи, М. М. Савченко

Національний технічний університет України

«Київський політехнічний інститут імені Ігоря Сікорського»

Проспект Перемоги, 37, 03056 Київ, Україна

Засоби візуалізації класів і взаємозв'язків об'єктно-реляційної бази даних Caché

Статтю присвячено аналізу засобів візуалізації складу класів, які визначено в об'єктно-реляційній базі даних Caché корпорації InterSystems. Проведено порівняння різних підходів до відображення класів Caché та визначено недоліки існуючих засобів візуалізації. Крім того, виявлено певну недостатність традиційної UML-діаграми класів для представлення складових класів Caché. Запропоновано відповідне розширення нотації UML і спосіб для найбільш інформативної візуалізації класів Caché, який забезпечує відображення елементів, що характерні класам БД Caché. Наведено опис веб-інтерфейсу візуалізації.

Ключові слова: візуалізація, InterSystems Caché, діаграма класів.

Вступ

Об'єктно-реляційна база даних InterSystems Caché — це одна з постреляційних баз даних, яка поєднує можливості традиційних реляційних баз даних та об'єктно-орієнтованих баз даних.

Логічна модель збереження даних, яка застосована в Caché, являє собою ієрархічне дерево. Відомо, що будь-яке логічне дерево представляється у вигляді вузлів і даних. Кожен вузол має свій власний ідентифікатор, унікальний на своєму логічному рівні у списку нащадків свого батьківського вузла. Кожен вузол існує в разі, коли у нього є дані (визначено значення вузла), або є нащадки, або якщо створені і дані, і нащадки. Повний шлях або ідентифікація вузла визначається за допомогою перерахування ідентифікаторів усіх його предків. База даних Caché являє собою набір таких логічних дерев. Логічне дерево, яке зберігається в БД Caché, прийнято називати глобальною. Caché реалізує на базі ієрархічної моделі і реляційну, і об'єктну моделі доступу до даних. Програмні засоби сервера бази даних Caché [3], на основі використання ієрархічної моделі забезпечують для користувача підтримку можливості застосовувати наступні засоби доступу до інформації:

- прямий доступ до глобалей (створення, модифікація, читання);
- об'єктний доступ на основі використання об'єктної моделі (створення класів та об'єктів, їхнє збереження та використання);
- реляційний доступ, шляхом застосування вбудованого механізму обробки SQL-запитів (традиційні SQL-запити для обробки відношень) на основі використання реляційної моделі.

Таким чином, Caché — це ідеальна машина бази даних, в якій можна побудувати практично будь-яку логічну модель даних.

Під час проектування та розробки інформаційних систем із застосуванням об'єктно-орієнтованих технологій формується представлення предметної області у вигляді сукупності класів, які відображають основні сутності предметної області. У подальшому відбувається формування множин екземплярів визначених класів, і виконуються операції над ними, які дозволяють реалізувати функціональні задачі системи, що проектується.

Існує багато інструментальних засобів, які призначені для здійснення об'єктно-орієнтованого підходу до проектування, переважна більшість яких передбачає застосування нотації уніфікованої графічної мови моделювання предметної області — мови UML [3]. Проект системи у цьому випадку являє собою сукупність діаграм, серед яких важливе місце відведено діаграмі класів, яка наочно представляє користувачеві визначену сукупність класів і їхню структуру.

Для збереження інформації користувача можна застосувати об'єктно-орієнтовану базу даних Caché корпорації InterSystems, яка надає користувачу інтерактивні інструменти для формування потрібних класів достатньо складної структури, у зв'язку з цим, виникає задача формування засобів зручної візуалізації структури створених класів, з метою проведення її аналізу та удосконалення для застосування у нових проектах.

Саме засобам візуалізації структури створених класів ООБД Caché і присвячена дана стаття.

Актуальність

Останнім часом розробники інформаційних систем все більше і більше схиляються до використання об'єктно-реляційних баз даних для зберігання та обробки інформації, оскільки такі бази даних поєднують переваги реляційного та об'єктного доступу до інформації. Серед таких СКБД привертає особливу увагу об'єктно-реляційна база даних Caché корпорації InterSystems, яка надає додаткову можливість використання прямого доступу до інформації з використанням внутрішнього формату збереження даних у вигляді багатовимірних масивів [1]. Особливу увагу користувачів привернула реалізація об'єктного доступу, який найбільше відповідає уявленню користувача про інформаційну структуру предметної області. Архітектура класів об'єктно-реляційної бази даних InterSystems Caché, на відміну від усіх інших СКБД, має велику кількість особливостей, тому наявність засобів зручного представлення класів для користувача, який є розробником системи стає край актуальною задачею. Існуючі рішення візуалізації класів і відношень між класами, що зазвичай основані на нотації UML, не дозволяють повністю відоб-

разити всі особливості класів InterSystems Caché для отримання повноцінної візуалізації, яка б не приховувала важливих для даної СКБД деталей. У зв'язку з наведеним, постає необхідність розширити існуючий стандарт візуалізації UML, доповнивши його необхідними деталями та особливостями, яких у ньому не вис-тачає, та розробити відповідні програмні засоби підтримки.

Постановка задачі

СКБД Caché включає набір засобів і інструментів, які дозволяють переглядати класи та їхні елементи кількома способами. Основним засобом визначення сукупності створених класів предметної області є використання візуального ін-терактивного середовища розробки — студії Caché. Студія надає користувачеві декілька допоміжних діалогів (візардів), за допомогою яких можна визначити клас та його параметри, визначити властивості, методи та запити класу. Допоміж-ні діалоги на основі інтерактивного спілкування з користувачем створюють від-повідний код визначення класу та його складових внутрішньою мовою програму-вання СКБД Caché — Caché ObjectScript (COS). Професійні програмісти також мають можливість визначити клас шляхом безпосереднього формування програм-ного коду. Студія Caché надає користувачеві можливість перегляду створеного програмного коду визначення класу. Цей спосіб дозволяє отримати всю необхідну інформацію про клас, його властивості та взаємозв'язки з іншими класами.

Повнота інформації, яку може отримати користувач, якій не обов'язково є професійним програмістом, під час перегляду програмного коду класів напряду залежить від його вміння розуміти створений код. У багатьох випадках знайти не-обхідну інформацію про клас потрібно тим користувачам, які не працюють тісно з Caché ObjectScript. Синтаксис мови COS достатньо специфічний, і прочитати, а головне, правильно сприйняти програмний код буває досить складно.

Найбільш простий для сприйняття людиною спосіб подання даних — це на-очна візуалізація інформації, наприклад, за допомогою інфографіки [3]. На відмі-ну від текстового подання інформації, візуальні образи швидко сприймаються і часто краще запам'ятовуються. Виходячи із цих тверджень, поставлено задачу розробки інструмента візуалізації сукупності класів з метою забезпечення аналізу існуючих систем і проектування нових, який би дозволив показати усі особливос-ті та характеристики створених класів, не втрачаючи при цьому інформативності порівняно з переглядом програмного коду класу.

Огляд наявних способів відображення класів і їхніх характеристик

Інтегрована середа розробки, що поставляється разом із Caché, дозволяє пе-реглядати класи кількома способами, а саме:

- перегляд визначення класу у вигляді рядків тексту;
- перегляд за допомогою оглядача класів;
- перегляд за допомогою інспектора елементів проекту.

Текстове представлення визначення класу, в якому, наприклад, створені дві властивості та один індекс, виглядає наступним чином:

```

Class MyApp.Person Extends %Persistent [ClassType = persistent]
{
Property Name As %String;
Index NameIdx On Name;
Property Age As %Integer;
}

```

Оглядач класів та інспектор об'єктів є складовими частинами студії розробки [4]. Оглядач надає користувачеві представлення структури класу у вигляді ієрархічного дерева, що складається з певного набору гілок (рис. 1). У режимі перегляду властивостей класу відображаються такі дані, як, наприклад, «назва властивості» і «тип», а при перегляді методів — «назва методу» та «тип даних, що повертаються». Звісно, такий спосіб подачі інформації є досить інформативним, але він вимагає формування великої кількості колонок у разі, якщо певна властивість має свої особливі параметри [3]. Це значно збільшує обсяг таблиць та час, який необхідно буде витратити на сприйняття та пошук інформації.

Інспектор елементів, що по замовчуванню знаходиться в правій частині інтерфейсу студії, дозволяє переглянути атрибути властивостей у більш зручному вигляді (рис. 2). Під час перегляду списку властивостей, який представлений таблицею, можна двічі натиснути на певну властивість і відкриється таблиця з усіма параметрами обраної властивості, яка міститиме усього два стовпчики — назву параметра та його значення.

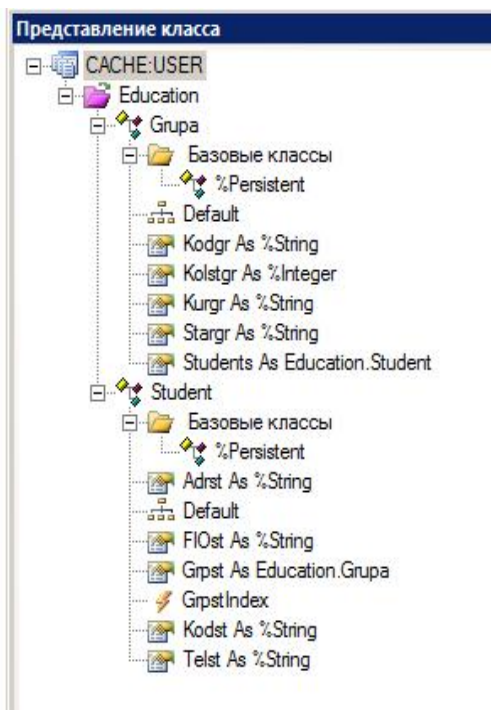


Рис. 1. Вікно оглядача класів

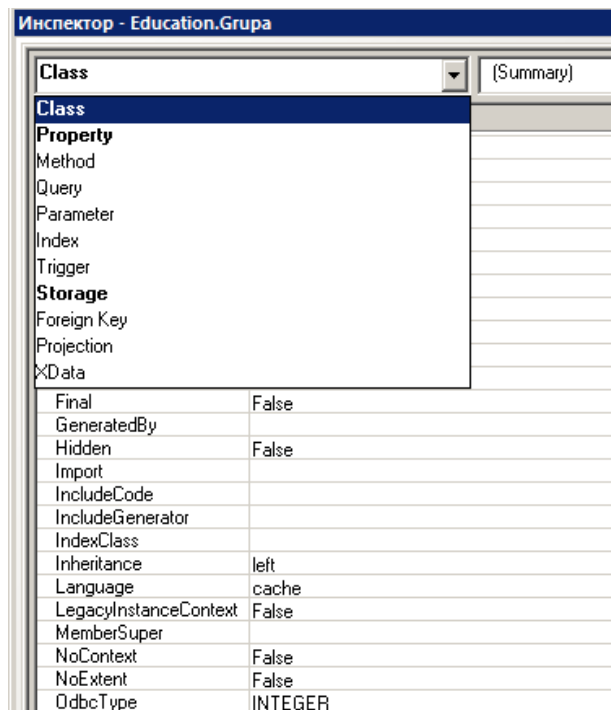


Рис. 2. Вікно інспектора елементів

Таким чином, табличне представлення складових класу — методів, властивостей та інших сутностей — вимагає використання різних типів візуальних таб-

лиць, між якими потрібно перемикається. Такий спосіб не дозволяє переглянути всю інформацію одразу, щоб максимально швидко скласти загальне враження про клас, що розглядається. Також часто наявність будь-яких зв'язків між класами є зовсім не очевидною до того моменту, доки користувач не перегляне усі необхідні класи, уявляючи при цьому схему зв'язків, і не складе всю отриману інформацію в одну цілісну картину.

У рамках поставленої задачі найбільш сприятливою формою візуалізації схеми класів розглядається нотація Unified Modeling Language (UML), яка є загально визначеним засобом для об'єктно-орієнтованого моделювання в області програмного забезпечення, моделювання бізнес-процесів, системного проектування тощо [5].

Стандартна діаграма класів у нотації UML дозволяє коротко описати всі властивості та методи класу, а також показати взаємозв'язки між ними — асоціації, наслідування, та, що характерне для сховища даних — різні зв'язки цілісності даних. Клас представляється у вигляді прямокутника, поділеного на секції — в одній знаходяться методи, у іншій — властивості класу. У випадку з властивостями, кожен рядок у відповідному прямокутнику містить назву властивості та її тип.

Спроба застосування стандартної діаграми класів до відображення повної структури класів СКБД Caché, яким притаманна, з одного боку, наявність нетипових складових (параметри, запити, відношення), а з іншого — поширена інформація про самі складові (такі як, наприклад, параметри властивостей) виявляє її недосконалість, тому що в ній відсутні відповідні можливості.

Ситуація, що склалася, призвела до необхідності розробки та застосування розширеної нотації UML і створення додаткових візуальних позначок, які би дозволили стисло та максимально повно передати всю необхідну інформацію про клас.

Клас у Caché, окрім методів і властивостей, містить також інші особливі сутності. Такими сутностями є: параметри — спеціальні незмінювані властивості, які характеризують всі екземпляри класів; запити — процедури, які повертають заздалегідь визначений набір даних за певними критеріями; xData-блоки, що містять код у форматі XML; індекси, що дозволяють підтримувати цілісність даних і пришвидшити доступ до них, а також два особливі типи зв'язків, окрім наслідування й асоціації, — зв'язки «One-to-Many» та «Parent-Child».

Перші три із вищезазначених сутностей можна виділити в такі ж логічні блоки, якими розділяються методи та властивості у нотації UML, тому що і параметри, і запити, і xData-блоки, подібно до методів і властивостей, мають свою назву та тип. У такому разі клас на діаграмі буде розділено вже на п'ять логічних блоків — параметри, властивості, методи, запити та xData-блоки класу. Для класів, які зберігаються в базі даних, останні три блоки іноді залишатимуться пустими. У такому разі ці блоки не варто відображати взагалі, а для всіх видимих блоків додати заголовки, який показував би, сутності якого типу відображаються у даному блоці.

Кожна складова класу Caché (властивість, метод та інші) має також і свої власні параметри, які певним чином характеризують цю складову. Наприклад, окрему властивість можна зробити «фінальною» — тобто заборонити її перевизначення при наслідуванні. Для цього до цієї властивості потрібно додати пара-

метр Final. Кількість таких параметрів, що певним чином змінюють характеристики сутності, досить невелика, але вони часто вживаються.

Для декількох найуживаніших параметрів, таких як Private, Final, Abstract та інших, пропонується додати графічні іконки, що будуть відображатися ліворуч від назви сутності та дозволяти користувачеві за зображенням визначити, якими основними параметрами визначається та чи інша сутність.

Одночасно потрібно врахувати те, що кількість інформації, яка буде виводиться графічно, все ж таки має бути мінімізованою задля збереження компактного вигляду діаграми. Тому всі додаткові відомості про методи, властивості, параметри класу, запити та xData-блоки, такі як коментарі, документація, повний перелік параметрів сутності та її детальний опис можна приховати та застосувати для них засоби інтерактивності графічного середовища відображення, які надають можливість виводити детальну інформацію за бажанням користувача. Наприклад, якщо користувачеві знадобиться додаткова інформація про властивість класу, він зможе навести на неї курсор миші та отримати цю інформацію у контекстному вікні. Приклади реалізації запропонованого рішення наведені на наступних рисунках: типове відображення структури окремого класу (рис. 3), відображення структури зв'язаних класів (рис. 4) та відображення фрагменту концептуальної схеми бази даних (рис. 5).

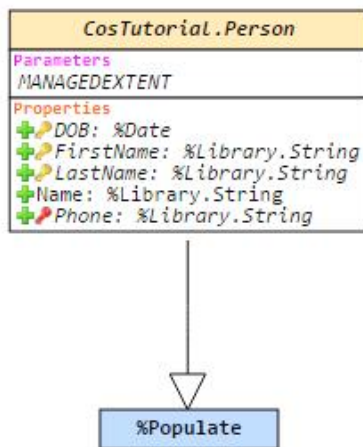


Рис. 3. Типове відображення структури зв'язаних класів

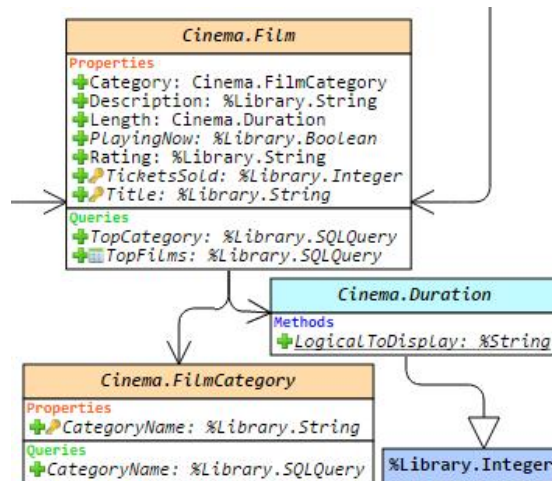


Рис. 4. Відображення структури окремого класу

У загальному випадку, клас, як сутність, має свій певний набір параметрів. Але є особливий параметр, який визначає тип класу — ClassType. Тип класу несе за собою базову характеристику класу, яка визначає, чи буде він зберігатись у базі даних (Persistent), буде зберігатися як частина існуючого класу, яка виконує окремі функції (Serial), або буде узагальнюючим типом даних для класів, що реєструються у системі (Registered). Окрім графічних іконок, що будуть відображати всі інші параметри класу, важливим для сприйняття користувачем є саме тип класу, і щоб показати його як головну характеристику, пропонується визначити окремий колір для кожного типу класу, відображаючи заголовки класу відповідними кольорами.

Класи, між якими встановлено зв'язки підтримання цілісності даних (відношення) «One-to-Many» та «Parent-Child» пропонується з'єднувати нерозривними лініями, на кінцях яких біля відповідних класів будуть розташовані написи «one» та «many» для зв'язків «One-to-Many» та «parent» і «child» для зв'язків «Parent-Child». Інтерактивність графічного середовища у такому випадку допоможе краще зрозуміти, які властивості були використані для встановлення зв'язку — наведення курсору миші на лінію зв'язку підсвічуватиме відповідні властивості.

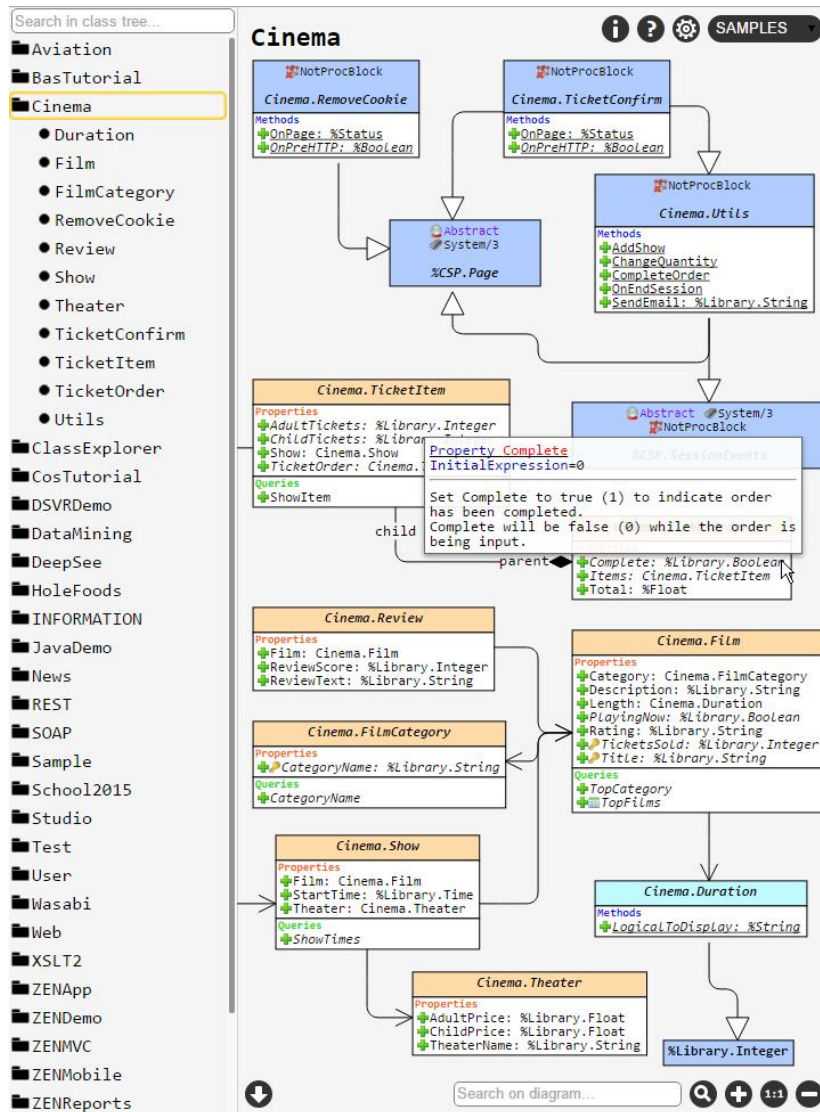


Рис. 5. Відображення фрагменту концептуальної схеми бази даних

Інструмент для відображення діаграми класів Caché розроблений у вигляді веб-застосунку, який може бути запущений у будь-якому сучасному веб-браузері, в тому числі і на мобільних платформах [6]. Для роботи серверної частини інструмента необхідно провести інсталяцію ООБД Caché з веб-сервером Caché, що підтримує стандарт REST. Інструмент поставляється у вигляді XML-файлу, який

потрібно імпортувати в Caché та скомпілювати. В результаті компіляції у Caché з'явиться новий веб-застосунок «/ClassExplorer/», який можна буде відкрити через веб-браузер.

Розроблена модифікація стандарту UML є унікальною. Авторами розглянуто декілька функціонально аналогічних програмних продуктів, зокрема:

1) StarUML — програмна платформа моделювання, яка підтримує UML, що заснований на версії UML 1.4 і підтримує нотацію UML версії 2.0 та одинадцять різних типів діаграм, підхід MDA (Архітектура Керована Моделлю) і концепцію профілів UML;

2) CASE Studio 2 — інструмент розробки баз даних, якій підтримує базу даних PostgreSQL. Для нової версії розроблено понад двадцять поліпшень і запропоновано нові функції, такі як повна підтримка бази даних MySQL 5.0;

3) PowerBuilder — інтегроване середовище розробки додатків баз даних з власною скриптовою мовою PowerScript — мовою програмування четвертого покоління, розробленим фірмою PowerSoft.

На момент розробки жоден з них не реалізував схожих інструментів для візуалізації класів СКБД Caché з усіма їхніми особливостями. Існуючі рішення до моменту розробки Caché Class Explorer дозволяли будувати лише обмежені, стандартні UML-діаграми, які не містили повної інформації про класи, тобто приховували її.

Висновок

Розроблено веб-систему візуалізації класів і їхньої структури об'єктно-орієнтованої бази даних Caché у вигляді удосконаленої UML-діаграми класів, з урахуванням особливостей побудови класів Caché. На діаграмі присутні тільки основні, найбільш важливі елементи, а за необхідності отримати будь-яку додаткову інформацію, таку як коментар або набір параметрів певного елемента, достатньо буде навести курсор миші на цей елемент.

За допомогою запропонованих підходів до графічного представлення інформації можна провести ефективний аналіз сукупності класів певної системи без необхідності вивчати її програмний код.

1. Кирстен В., Ирингер М., Рериг Б., Шульте П. СУБД Caché: Объектно-ориентированная разработка приложений. Санкт-Петербург: Питер, 2001. 384 с.

2. Шевченко В. Візуалізація інформації в ЗМІ. *Світ соціальних комунікацій. Науковий журнал*. 2012. Т. 7. С. 78–82.

3. OMG/UML, UML 2.1. Superstructure Specification, Document: ptc/06-04-02, 2006.

4. Гайдаржи В.І., Михайлова І.Ю. Об'єктно-реляційна СУБД Caché. Київ: Освіта України, 2015. 312 с.

5. Рамбо Дж., Блах М. UML 2.0. Об'єктно-орієнтоване моделювання і розробка. Вид-во «Пітер», 2-ге вид., Пітер, 2007.

6. Савченко М.М. Caché Class Explorer. 2016. URL: <https://github.com/intersystems-community/ClassExplorer>

Надійшла до редакції 01.11.2017