

КОМПОНЕНТНО-ОРІЄНТОВАНИЙ ПІДХІД ДО РОЗРОБКИ ПРОГРАМНИХ СИСТЕМ НА ОСНОВІ ІНТЕГРАЦІЇ МЕНЕДЖЕРІВ ІНФОРМАЦІЙНИХ РЕСУРСІВ

Розглядається метод побудови програмних систем на основі компонентної інтеграції менеджерів інформаційних ресурсів. У рамках методу визначається базова термінологія, типова схема з'єднання інтегрованого середовища та менеджерів ресурсів, структура менеджера, типові інтерфейси щодо реалізації доступу до ресурсів. Наведено опис застосування інтеграції Web-ресурсів на основі запропонованого методу.

Вступ

Сучасні інформаційно-обчислювальні комп'ютерні системи стають все більш складними об'єктами, що спричиняє постійне вдосконалення існуючих методів розробки та пошук нових концепцій, підходів, засобів для побудови таких систем. Значне місце у цій проблематиці посідають проблеми інтеграції програмних елементів, об'єктів даних, різнорідних інформаційних ресурсів тощо.

Наявний рівень проблем інтеграції насамперед пов'язаний з необхідністю вирішення задач забезпечення взаємодії широкого класу застосувань на основі нових та вже існуючих програмних елементів, побудованих на різних платформах, розроблених згідно з різними методологіями програмування, створених у різні періоди розвитку комп'ютерних та інформаційних технологій.

Вирішення відповідних проблем базуються на сучасних досягненнях теорії та практики програмування, у тому числі на результатах компонентного та об'єктно-орієнтованого програмування, на нових підходах до розробки розподілених систем, зокрема технологіях Web-сервісів, на методах повторного застосування

існуючих компонентів та інформаційних систем.

Одним з найбільш перспективних підходів є побудова програмних систем на основі інтеграції програм, яка базується на концепції менеджерів інформаційних ресурсів.

Сутність та приклади реалізації підходів до застосування концепції менеджера інформаційних ресурсів

Сутність концепції побудови програмних систем на основі інтеграції менеджерів ресурсів полягає у тому, що розробник окремої програми чи підсистеми разом із засобами, що безпосередньо реалізують основну функціональність, створює спеціальний програмний елемент – менеджер ресурсів, за допомогою якого і відбувається взаємодія.

На властивості менеджера ресурсів впливають багато факторів, головними з них є:

- цільове середовище, у рамках якого інтегруються програми та підсистеми;
- механізми встановлення з'єднань та організації взаємодії;
- наявність типових та стандартизованих прикладних інтерфейсів (API);
- способи та методи подання даних для роботи з менеджером ресурсів;
- забезпечення підтримки протоколів взаємодії та загальносистемних сервісів, на основі яких функціонує цільове середовище;
- рівень уніфікації та типізації інформаційних ресурсів, що інтегруються у середовище.

Залежно від вибору основних впливаючих факторів та рівня вирішення відповідних проблем існують багато концепцій застосування менеджерів інформаційних ресурсів. Наведемо кілька найбільш поширених прикладів.

Типовим прикладом архітектурних та структурних рішень на основі застосування менеджерів ресурсів є концепція ODBC щодо організації доступу до баз даних (БД) [1]. Основою цієї концепції є відповідний стандарт, що визначає архітектуру та принципи реалізації програмних засобів організації взаємодії з БД у мережевому середовищі. Типовим клієнтом в цій архітектурі є певне застосування, яке за допомогою стандартного прикладного інтерфейсу (ODBC API) звертається до спеціального менеджера – ODBC Driver Manager. Цей менеджер є типовим елементом, оскільки реалізує ODBC API для різних систем керування базами даних (СКБД) на рівні типових функцій та структур даних. Для звертання до конкретної БД ODBC Driver Manager застосовує відповідний драйвер ODBC, який враховує її особливості. Архітектура, побудована на концепції ODBC, зображена на рис. 1.

Фактично ODBC Driver Manager разом з ODBC-драйвером за своєю сутністю є менеджером ресурсу для відповідної БД.

Іншим прикладом схожої архітектури є концепція JDBC [2]. Вона в цілому еквівалентна архітектурі ODBC, проте застосовується в середовищі Java-технологій доступу до реляційних БД. Існують реалізації JDBC, створені на основі драйверів ODBC.

Провайдери даних у технологіях OLE DB [1] або ADO.NET [3] також є прикладами менеджерів ресурсів для доступу до БД, документів, файлів і т. д.

Найбільш багатоаспектним прикладом реалізації концепції менеджера ресурсів є архітектура конектора, яка є складовою Java-середовища J2EE [4]. Головна мета цієї архітектури – забезпечити уніфікований підхід до інтеграції різних програмних засобів та систем на основі типових механізмів взаємодії, стандартних інтерфейсів та сервісів, єдиних концепцій побудови інтегрованого середовища. Сферою застосування є корпоративні системи, що складаються з різноманітних програмних засобів та об'єктів, які охоплюють існуючі корпоративні інформаційні ресурси, створені у різний час та на різних платформах.

Для кожного інформаційного ресурсу необхідно існування спеціальної підсистеми доступу до нього, принципи побудови та функціонування якої визначаються специфікацією архітектури конектора. Специфікація визначає також такі поняття, як ресурс корпоративної інформаційної системи

(EIS Resource), менеджер ресурсів (Resource Manager), адаптер ресурсів (Resource Adapter), системний та прикладний контракти (System and Application Contracts) та ін. Особливостями згаданої архітектури є специфікація таких елементів, як забезпечення умов відповідності адаптера ресурсів вимогам загальних системних сервісів (з'єднання, транзакцій, безпеки), життєвого циклу адаптера ресурсів, інтерфейсів керування з'єднаннями, транзакціями, безпекою, роботами та ін.

Загальна схема побудови середовища на основі інтеграції ресурсів корпоративної системи наведена на рис. 2. Ядром інтеграції є сервер застосувань, який функціонує як J2EE-сервер. Він побудований на основі Java-технологій, що забезпечують підтримку стандартних механізмів взаємодії Java-об'єктів та реалізацію основних загальносистемних сервісів, таких, як іменування, транзакцій, безпеки та ін. Зокрема, сервер застосувань має спеціальне розширення, що підтримує реалізацію певних API, які забезпечують стандартні методи роботи з адаптерами інформаційних

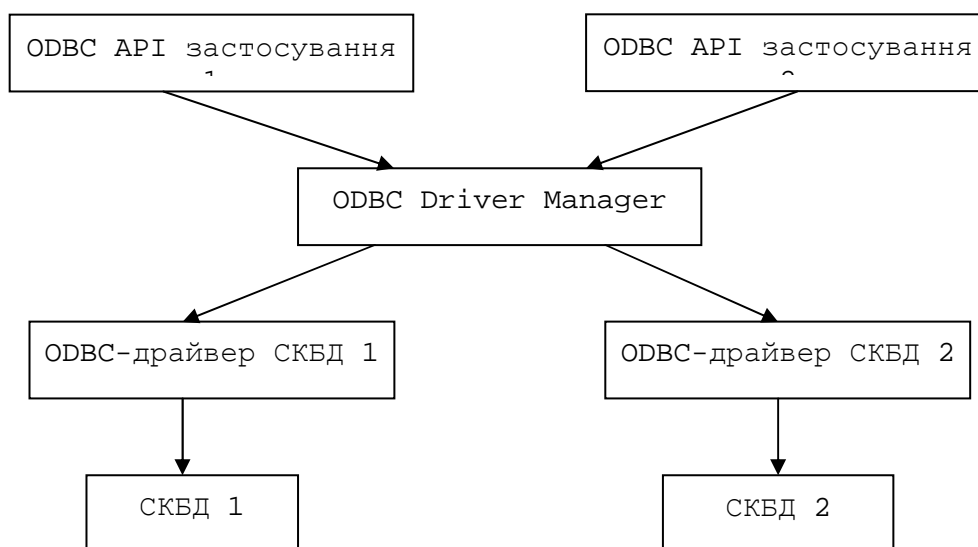


Рис. 1. Архітектура ODBC-доступу до БД

ресурсів.

Крім того, адаптери, що безпосередньо взаємодіють з ресурсами, також повинні задовольняти вимогам їх інтерфейсів, а також умовам підтримки загальносистемних сервісів. Стандартизація механізмів взаємодії дозволяє інтегрувати з одним сервером застосувань кільком корпоративним інформаційним ресурсам, а також інтегрувати певний ресурс з кількома J2EE-серверами.

Концепція методу побудови програмних систем на основі компонентної інтеграції менеджерів ресурсів

Існує багато реалізацій підходу до інтеграції менеджерів ресурсів у інтегрованих середовищах. Кожна з них має свої особливості, виходячи з вирішення певних завдань, вибору базових об'єктів для інтеграції, способів та механізмів взаємодії тощо. Сфера застосування кожного підходу залежить від рівня уніфікації, типізації, стандартизації окремих аспектів та складових цього підходу. Наприклад, реалізація концепції ODBC націлена на вирішення проблеми інтеграції баз даних і сфера її застосування цілком визначається метою створення відповідної архітектури.

Крім відмінностей у цілях та завданнях, окремі підходи до інтеграції можуть мати різні теоретичні та методичні засади, бути побудовані із застосуванням різних моделей та платформ, функціонувати на основі різних механізмів та технологій тощо. Вище наведені основні фактори, які впливають на особливості мене-

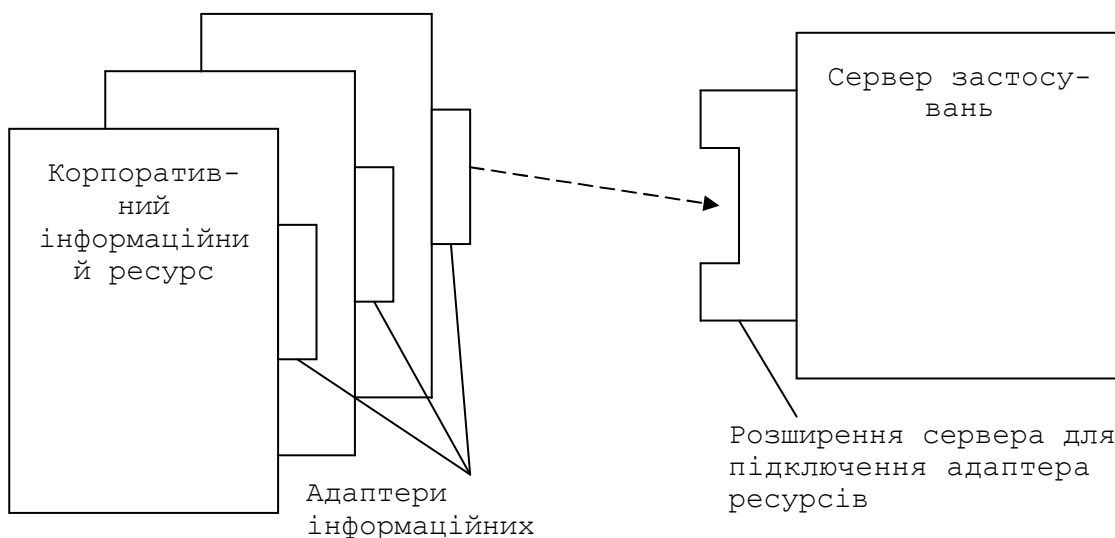


Рис. 2. Загальна схема інтеграції інформаційних ресурсів в архітектурі конектора

джерів ресурсів. Значним чином те ж саме впливає на конкретні реалізації підходів щодо інтеграції. Деталізація впливу та властивості цих факторів на процес інтеграції визначає певний метод інтеграції на основі менеджерів ресурсів.

Різноманітність у підходах та реалізаціях завдає певні труднощі у розповсюдженні та розширенні сфери застосування концепції інтеграції менеджерів ресурсів. Різні середовища інтеграції, різні типи прикладних інтерфейсів, різні правила взаємодії – все це обмежує можливість інтеграції різнорідних ресурсів. Наприклад, якщо до складу цільового середовища входять різні типи серверів інтеграції, то для кожного інформаційного ресурсу повинні бути розроблені кілька типів менеджерів ресурсів, що значно ускладнює застосування згаданої концепції, збільшує обсяг робіт щодо програмування, ускладнює правила створення інтегрованого середовища та взаємодії його окремих елементів.

Виходом з цієї ситуації є застосування таких теоретичних та практичних засад, які б найбільшим чином сприяли вирішенню проблеми, а також найбільш широко охоплювали різноманітні аспекти концепції для забезпечення необхідного рівня уніфікації та стандартизації структури середовища, правил інтеграції, прикладних інтерфейсів та ін. Одним з перспективніших є компонентно-орієнтований підхід, який забезпечує відповіді на всі головні питання практичної реалізації концепції інтеграції менеджерів інформаційних ресурсів на єдиній теоретичній, методичній та прикладній основі.

У цій статті пропонується метод побудови програмних систем на основі компонентної інтеграції менеджерів ресурсів. Компонентна орієнтація фіксує та визначає багато аспектів інтеграції, основними з яких є:

- всі об'єкти інтеграції – суть компоненти, у тому числі і менеджери інформаційних ресурсів;
- основа інтегрованого середовища – це компонентний

framework [5];

- сервери застосувань та їх загальносистемні сервіси повинні забезпечити виконання необхідних операцій щодо розгортання, функціонування, взаємодії програмних компонентів [2];

- механізмом взаємодії в інтегрованому середовищі є взаємодія компонентів на основі їх інтерфейсів [6];

- менеджери інформаційних ресурсів інкапсулюють самі ресурси від інтегрованого середовища і у цій якості виступають як узагальнені інтерфейси;

- кожен компонент має типову структуру, яка визначає його властивості, механізми функціонування та життєвий цикл (це, зокрема, наявність контейнера та компонентних реалізацій, існування механізму створення екземплярів компонента на основі фабрики компонентів, схема взаємодії компонентів на основі інтерфейсів та загальносистемних сервісів і т. д.) [6].

Основні поняття та визначення

Інформаційний ресурс – це логічно впорядкована сукупність даних та засобів їх подання, яка забезпечує створення, збереження та застосування інформації про визначені предметні та проблемні галузі, окремі аспекти та властивості, колекції об'єктів даних, кількісні та якісні характеристики та показники тощо. Це визначення носить узагальнений характер і забезпечує широкий спектр інтерпретації ресурсів як складових інформаційно-обчислювальних систем. Головним призначенням ресурсу є постачання інформації клієнтові згідно його запиту. Механізми створення, формування даних, а також їх збереження, подання інкапсульовані у ресурсі. Це означає, що способи формування інформації можуть носити довільний характер. Головним є те, щоб у відповідь на свій запит клієнт ресурсу отримав

необхідні дані. Зокрема, ресурсом може бути прикладна програма, яка генерує вихідну інформацію, що сприймається ззовні як дані з ресурсу.

Класифікація інформаційних ресурсів – сукупність принципів типізації та впорядкування множини інформаційних ресурсів згідно їх функціональності, методів і засобів подання, алгоритмів, схем і механізмів отримання даних, доступу до інформації. Відповідно до введеного визначення інформаційних ресурсів існує багато класифікацій. Для проблем інтеграції найбільш поширеними є:

- за типами об'єктів даних (документи, таблиці, колекції тощо);

- за структурними ознаками (файли, каталоги, розділи і т. д.);

- за методами формування (бази даних, Web-сайти, прикладні програми та застосування тощо).

Тип інформаційного ресурсу

– клас інформаційних ресурсів, визначений конкретними значеннями класифікаційних ознак відносно певної їх класифікації. Стосовно проблем інтеграції типізація інформаційних ресурсів визначає методи та механізми реалізації ресурсу, подання інформації, а також доступу до неї. Прикладом є реляційні БД, де подання даних здійснюється на основі реляційної моделі, механізмом реалізації ресурсу є реляційна СКБД, а методами доступу до інформації – мова SQL.

Іншим прикладом може бути файл даних у форматі XML. Подання таких даних засноване на мові розмітки XML. Засобом створення XML-файлу може бути відповідний генератор або редактор XML-даних, а механізми доступу побудовані на спеціальних засобах, які називаються парсерами.

Головне призначення типізації інформаційних ресурсів поля-

гає в уніфікації і стандартизації методів створення, подання та застосування даних для певної підмножини інформаційних ресурсів виходячи з їх функціональної сутності, технічних і технологічних особливостей.

Менеджер інформаційного ресурсу – програмний об'єкт інформаційно-обчислювальної системи, який забезпечує керування та його застосування відповідним інформаційним ресурсом на основі стандартизованих правил і механізмів доступу до даних ресурсу. Менеджер інформаційного ресурсу за своєю сутністю є оболонкою над ресурсом, що реалізує стандартизовані інтерфейси для забезпечення процесів інтеграції.

Тип менеджера інформаційного ресурсу – клас програмних об'єктів, який визначається сукупністю уніфікованих і стандартизованих методів застосування та доступу до ресурсів певного типу. В окремому випадку певний тип може складатися тільки з одного екземпляра. Така ситуація часто має місце, коли до складу інтегрованого середовища залучається існуюча система, створена, наприклад, на старих платформах і мовах програмування. Тоді розробляється менеджер ресурсів для забезпечення взаємодії тільки з такою системою.

Головна мета типізації менеджерів інформаційних ресурсів полягає у впорядкуванні та уніфікації всіх інтерфейсів, які з'єднують менеджера з певним класом інформаційних ресурсів та з інтегрованим середовищем. Прикладом, який ілюструє сутність типізації, є менеджер роботи з XML-файлами для об'єктно-орієнтованої програмної системи. Такий менеджер взаємодіє з ресурсом на основі методів і засобів для мови XML. З боку інтегрованого середовища існує вимога подання даних з ресурсу як окремих об'єктів. Тому менеджер повинен дані, що відбираються з

XML-файлу, подавати у вигляді об'єктів, наприклад, на основі DOM-моделі [2].

Інтегроване середовище – сукупність програмних елементів, об'єктів даних і системних сервісів, які впорядковані відносно зв'язків між собою, взаємодіють один з одним на основі уніфікованих та стандартизованих механізмів і функціонують відповідно до єдиних правил побудови та еволюції, що визначають сутність середовища.

Інтегроване компонентне середовище – інтегроване середовище, де всі його складові подані як компоненти, механізми взаємодії, засновані на застосуванні інтерфейсів, а правила побудови та еволюції відповідають принципам компонентного програмування.

Типова схема з'єднання з менеджером інформаційних ресурсів

Типова схема з'єднання інтегрованого середовища з менеджером інформаційних ресурсів подана на рис. 3.

Уніфікований запит до менеджера ресурсу відображає структуру запиту з боку інтегрованого середовища до менеджерів ресурсів певного типу. Вище наведений приклад з архітектурою ODBC-доступу до БД, де структура уніфікованого запиту визначається ODBC API. За умов компонентної орієнтації уніфікований запит – це звернення до методу з вхідного інтерфейсу компонента-менеджера ресурсу.

Типізований запит до інформаційного ресурсу має структуру, яку безпосередньо розуміє конкретний інформаційний ресурс. Одна з функцій менеджера інформаційних ресурсів – трансформація уніфікованого запиту в один або послідовність типізованих запитів, які необхідно виконати, щоб отримати відповідну інформацію з ресурсу. Наприклад, уніфікований запит до БД трансформується у SQL-запит, щоб отримати інформацію

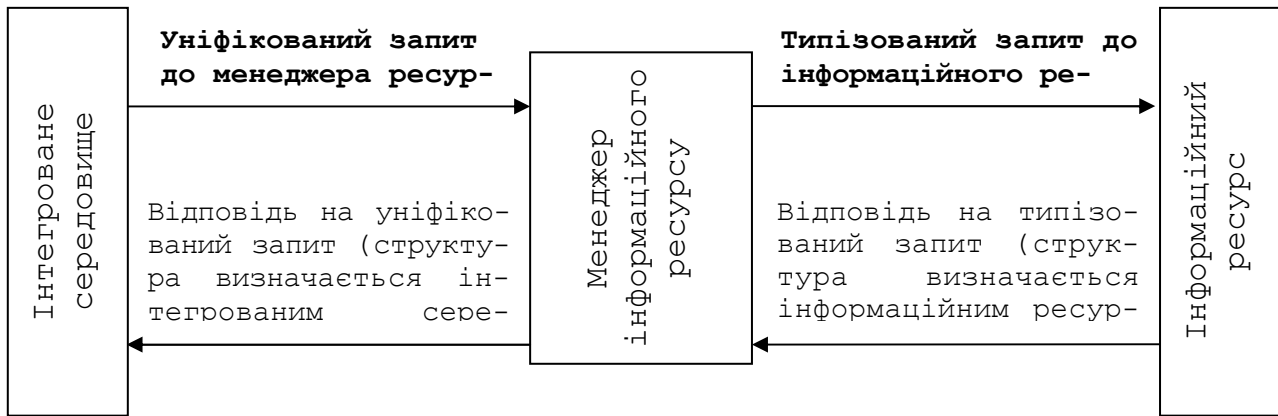


Рис 3. Типова схема з'єднання з менеджером інформаційних

цію з реляційної БД. Реалізація механізму типізованого запиту в цілому може відрізнятись від механізму компонентної взаємодії. Цей факт розглядається як внутрішній аспект у концепції подання менеджера ресурсів як узагальненого інтерфейсу, який інкапсулює внутрішні особливості реалізації.

Відповідь на типізований запит визначається структурою даних, яку формує інформаційний ресурс згідно зі своєю функціональністю. Наприклад, результат обробки SQL-запиту до БД подається у табличній формі (у випадку, коли ресурс містить необхідну інформацію).

Відповіддю на уніфікований запит є структура даних, стандартизована для певного інтегрованого середовища у якості типової відповіді для менеджера ресурсів. Наприклад, таблиця з попереднього прикладу може залишатись без змін або трансформуватись у XML-структуру даних, або бути подана як HTML-сторінка тощо. Механізм реалізації відповіді на уніфікований запит є складовою механізму компонентної взаємодії.

Доцільно провести порівняльний аналіз між схемою, що зображена на рис. 3, та компонентним середовищем. Основою останнього є компонентний framework, поданий як сукупність серверів компонентів, що взаємо-

діють між собою, а також сукупність компонентних сервісів [7, 8]. Між компонентами та компонентним framework існують стандартизовані механізми під'єднання компонента до середовища, організації взаємодії, обміну даними.

У запропонованому методі інтегрованим середовищем є компонентне середовище. Як наслідок, серверами інтеграції є компонентні сервери, а загальносистемні сервіси іменування, транзакцій, безпеки реалізуються для підтримки компонентів. Менеджер ресурсів – завжди компонент незалежно від методів побудови самого ресурсу (останній може бути реалізований як на компонентній основі, так і на довільній платформі).

Типова процедура включення ресурсу в інтегроване середовище містить наступні етапи:

- розгортання менеджера на одному з компонентних серверів згідно з загальними правилами розгортання компонентів;
- організація взаємодії менеджера з ресурсом відповідно до особливостей останнього (вибір адаптера або драйвера, зв'язок з місцезнаходженням ресурсу тощо);
- конфігурування менеджера у компонентному середовищі для забезпечення доступу з боку інших компонентів.

Типова структура менеджера інформаційного ресурсу

Менеджер інформаційних ресурсів – це підсистема, що складається з кількох компонентів. Типова структура менеджера зображена на рис. 4.

До типової структури входять наступні компоненти:

- контейнер, що створює та забезпечує керування екземплярами з'єднань з інформаційним ресурсом;
- екземпляри з'єднань з інформаційним ресурсом, що забезпечують реалізацію функціональності доступу до ресурсу;
- компонент керування взаємодією для певного з'єднання;
- адаптер доступу до ресурсу;
- дескриптор інформаційного ресурсу, що є інформаційною базою для створення з'єднання та організації взаємодії.

Ролі та функціональність між компонентами менеджера ресурсів розподілені наступним чином.

Дескриптор інформаційного ресурсу є інформаційною базою, що підтримує функціонування самого ресурсу та його менеджера протягом усього життєвого циклу. Зокрема, вона містить таку інформацію:

- опис параметрів розгортання менеджера як компонента;
- опис місцезнаходження або адреси ресурсу, а також параметрів доступу до нього;
- тип та характеристики менеджера ресурсу;
- опис інтерфейсів, методів, структур вхідних та вихідних параметрів;
- фрагменти документації менеджера та інформаційного ресурсу.

Менеджер, як типовий компонент, виступає у якості контейнера екземплярів, кожен з яких є реалізацією певного з'єднання з ресурсом. Екземпляр створюється під час ініціації з'єднання й існує до виклику методу закриття з'єднання з ресурсом.

Екземпляр з'єднання склада-

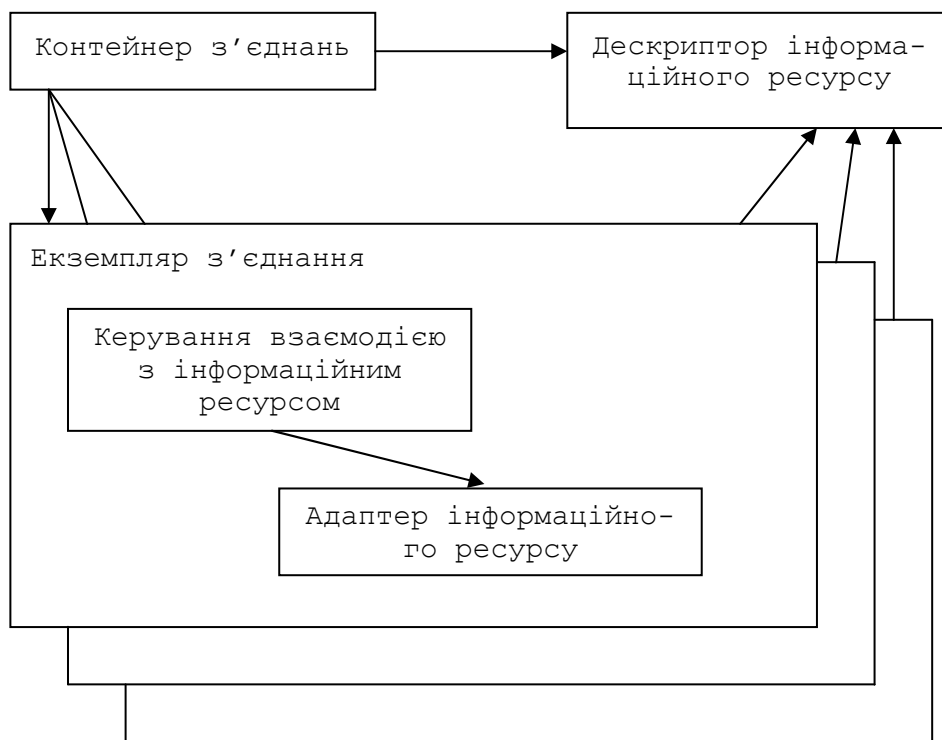


Рис. 4. Типова структура менеджера інформаційних ре-

ється з кількох компонентів, типовими з яких є компонент керування взаємодією з ресурсом та адаптер інформаційних ресурсів.

Компонент керування взаємодією з ресурсом забезпечує реалізацію вхідного інтерфейсу менеджера ресурсів, який визначає функціональність процедур доступу до ресурсу. До його функцій, зокрема, входять:

- перетворення вхідних параметрів уніфікованого запиту до менеджера у структури даних адаптера ресурсу;

- організація взаємодії з адаптером ресурсу;

- перетворення результатів виконання запиту у структури даних для вихідних параметрів.

В окремому випадку цей компонент може бути відсутнім, коли вхідні параметри уніфікованого запиту не потребують додаткової обробки і сприймаються адаптером ресурсу безпосереднім чином.

Адаптер ресурсу взаємодіє з інформаційним ресурсом відповідно до механізмів доступу, реалізованих у засобах керування ресурсом, наприклад засобах СКБД для доступу до БД.

Типові інтерфейси менеджера інформаційних ресурсів

У компонентній структурі взаємодія між усіма компонентами відбувається за допомогою інтерфейсів. Оскільки менеджер інформаційних ресурсів також подається як компонент, то його функціональність розглядається як реалізація певних інтерфейсів.

Відповідно до загальної архітектури, основних функцій та принципів побудови функціональності менеджера інформаційних ресурсів є реалізацією певної сукупності типових інтерфейсів:

- підтримки з'єднання з менеджером інформаційних ресурсів;

- забезпечення взаємодії з

менеджером;

- забезпечення подання даних.

Інтерфейс підтримки з'єднання з менеджером інформаційних ресурсів є інтерфейсом контейнера з'єднань відповідно до типової структури менеджера. Він забезпечує реалізацію наступних основних типових функцій:

- пошук необхідного менеджера в інтегрованому середовищі;

- створення екземпляра з'єднання для менеджера (одночасно з менеджером ресурсів можуть бути пов'язані декілька екземплярів з'єднань відповідно до кількості звертань до нього);

- встановлення системних контрактів, зокрема, для підтримки транзакцій та безпеки.

Інтерфейс забезпечення взаємодії з менеджером є прикладним інтерфейсом, що реалізує екземпляр з'єднання. Завдяки йому виконуються такі типові функції:

- отримання інформації про опис екземпляра з'єднання, зокрема переліку методів, які реалізують функціональність для отримання даних з ресурсу;

- звернення до методів для отримання необхідних даних з інформаційного ресурсу.

Інтерфейс забезпечення подання даних також є прикладним інтерфейсом, що реалізує екземпляр з'єднання. За його допомогою здійснюються:

- формування типових структур даних для інформації, отриманої з ресурсу;

- перетворення між різними структурами даних та їх поданням;

- операції з дескрипторами та метаданими для типових структур даних.

Для окремих реалізацій концепції інтеграції на основі менеджерів ресурсів можуть існува-

ти інші інтерфейси, які їм властиві.

Приклад реалізації Web-застосування на основі інтеграції менеджерів інформаційних ресурсів

Концепції та принципи створення інтегрованого середовища на основі інтеграції менеджерів ресурсів були використані при розробці Web-застосування для навігації та доступу до інформаційних ресурсів у мережі Інтернету. Програмні засоби засновані на компонентному підході до розробки Web-застосувань, який визначає, регламентує та підтримує виконання типових функцій та операцій у мережі Інтернету.

Основою інтегрованого середовища є Web-сервер, у рамках якого реалізовано компонентний framework, розроблений згідно з компонентним підходом [9]. Web-застосування для навігації та доступу до інформаційних ресурсів реалізоване як сукупність компонентів, що функціонують у серверному компонентному середовищі відповідно до правил та схем обробки даних, визначених у компонентному framework.

Інтегроване середовище містить три типи менеджерів інформаційних ресурсів:

- Web-сайтів;
- Web-застосувань;
- віддаленого доступу до

БД.

Структура уніфікованого запиту визначається стандартним для Web-середовища CGI-механізмом [10]. Аналогічно форматом відповіді на уніфікований запит є HTML-сторінка.

Структури типізованих запитів до інформаційних ресурсів та відповідей на них залежать від типу менеджера ресурсів.

Дескриптори ресурсів подаються як XML-файли, що визнача-

ють основні характеристики, за якими створюються відповідні з'єднання з ресурсами, та їх функціональність. Компонент, що забезпечує обробку даних з дескриптора, застосовує DOM – модель [2] для XML-файлів. Переліки всіх ресурсів, що складають загальну систему інформаційних ресурсів, також подані у вигляді XML-файлів, розподілених між кількома Web-серверами, на кожному з яких розгорнутий окремий екземпляр Web-застосування для навігації та доступу до інформаційних ресурсів. Кожен перелік визначає сукупність ресурсів, пов'язаних з конкретним сервером.

Менеджер ресурсів реалізовано як Java-компонент відповідно до загальної структури менеджера (див. рис. 4).

Менеджер ресурсів для Web-сайтів на основі уніфікованого запиту відшукує дескриптор ресурсу, за даними якого формує адресу цільового сайту та виконує переадресацію згідно сформованої адреси. Типовий дескриптор для цього типу менеджерів ресурсів має наступний вигляд:

```
<?xml version="1.0" encoding="WINDOWS-1251"?>
<ir_manager_description>
  <example_name>Web-сайт ІПС
  </example_name>
  <example_id>ir_manager_description_for_iss_site
  </example_id>
  <description>Web-сайт Інституту програмних
  систем НАНУ</description>
  <manager_type>web-site</manager_type>
  <location>
    <url>www.isofts.kiev.ua:80</url>
    <protocol>http</protocol>
  </location>
</ir_manager_description>
```

У цьому дескрипторі наведено найбільш важливі та найбільш вживані характеристики з опису ресурсу:

- ім'я;
- унікальний ідентифікатор у системі подання ресурсів;
- призначення;
- тип менеджера;
- адреса та протокол доступу до менеджера.

Менеджер ресурсів для Web-застосувань на основі уніфікованого запиту відшукує відповідний дескриптор ресурсу та опис методу доступу до нього. Метод доступу визначається сукупністю параметрів та їх значень. Після цього встановлюється адреса цільового сайту, де знаходиться Web-застосування, формується множина CGI-параметрів та їх значень, виконується переадресація згідно сформованої адреси. Приклад типового дескриптора для цього типу менеджерів ресурсів:

```
<?xml version="1.0" encoding="WINDOWS-1251"?>
<ir_manager_description>
  <example_name>Робота з Web-застосуванням</example_name>
  <example_id>ir_manager_description_for_app
  </example_id>
  <description>Моделює роботу з певним
  Web-застосуванням</description>
  <manager_type>web-application
  </manager_type>
  <location>
    <url>www.server.isofts.kiev.ua:
```

```
8080/app/test</url>
  <protocol>http</protocol>
</location>
  <request_list>
    <request>
      <description>Побудова дерева інформаційних
      ресурсів</description>
      <parameter_list>
        <parameter>
          <parameter_name>name
          </parameter_name>
          <parameter_value>*
          </parameter_value>
        </parameter>
        <parameter>
          <parameter_name>object
          </parameter_name>
          <parameter_value>tree
          </parameter_value>
        </parameter>
        <parameter>
          <parameter_name>type
          </parameter_name>
          <parameter_value>tree
          </parameter_value>
        </parameter>
      </parameter_list>
    </request>
    <request>
      <description>Пошук ресурсів відповідно
      пошуковому слову</description>
      <parameter_list>
        <parameter>
          <parameter_name>type
          </parameter_name>
          <parameter_value>search
          </parameter_value>
        </parameter>
        <parameter>
          <parameter_name>object
          </parameter_name>
          <parameter_value>*</parameter_value>
        </parameter>
      </parameter_list>
    </request>
  </request_list>
</ir_manager_description>
```

Це Web-застосування знаходиться на Web-сервері та реалізує дві прикладні функції: побудову дерева інформаційних ресурсів та пошук ресурсів за пошуковим словом, що входить до їх описів. Для кожної функції існує дескриптор запиту, який

визначає CGI-параметри та їх значення.

Менеджер ресурсів для віддаленого доступу до БД на основі уніфікованого запиту відшукує відповідний дескриптор ресурсу та опис методу доступу до ресурсу. Як і для Web-застосування, метод доступу визначається сукупністю параметрів та їх значень. На основі цих параметрів формується SQL-запит. Дескриптор ресурсу містить дані про JDBC-драйвер, який забезпечує зв'язок з ресурсом. Драйвер завантажується та ініціюється для доступу до відповідної БД. Сформований SQL-запит передається JDBC-драйверу, який організує його виконання. Результатом запиту є структура даних, яка описує таблицю з результатами обробки SQL-запиту. На основі цієї структури даних генерується фрагмент HTML-сторінки, яка посилається клієнтові. Приклад типового дескриптора для цього типу менеджерів ресурсів має наступний вид:

```
<?xml version="1.0" encoding="WINDOWS-1251"?>
<ir_manager_description>
  <example_name>Доступ до БД
  </example_name>
  <example_id>ir_manager_description_for_app_db
  </example_id>
  <description>Моделює роботу з доступу до баз даних</description>
  <manager_type>web-application-db
  </manager_type>
  <location>
    <url>localhost:8080/app/test</url>
    <protocol>http</protocol>
  </location>
  <db_connection_parameters>
    <driver>com.mysql.jdbc.Driver</driver>
    <url>jdbc:mysql://localhost:3306/test</url>
    <username>root</username>
    <password></password>
  </db_connection_parameters>
  <request_list>
    <request>
      <description>Запрос к БД</description>
      <parameter_list>
        <parameter>
          <parameter_name>type
```

```
</parameter_name>
        <parameter_value>db
      </parameter_value>
    </parameter>
  </parameter_list>
</request_list>
</ir_manager_description>
```

Структура дескриптора в цілому аналогічна наведеному для попереднього випадку. Додатково він описує параметри JDBC-драйвера для MySQL, тобто цільова БД як інформаційний ресурс створена та підтримується цією СКБД. При необхідності може бути застосована інша СКБД, наприклад Oracle. У цьому випадку потрібно змінити дані з дескриптора щодо JDBC-драйвера, який підтримує зв'язок з Oracle. Інші компоненти менеджера доступу до ресурсів змін не потребують.

Висновки

Проведено аналіз сутності та особливостей процесу створення програмних систем на основі концепції інтеграції менеджерів інформаційних ресурсів. Вказано на принципові обмеження існуючих підходів та реалізацій цієї концепції, а також обґрунтовані напрямки її вдосконалення. Головною метою при цьому є вибір та застосування єдиних теоретичних, методичних та практичних засад для вирішення проблеми.

У статті пропонується метод побудови програмних систем на основі компонентної інтеграції менеджерів ресурсів, тобто єдиною теоретичною та методичною

базою є компонентно-орієнтований підхід. Підґрунтя методу складає сукупність типових рішень щодо:

- структури інтегрованого середовища;
- схеми взаємодії середовища та менеджерів інформаційних ресурсів;
- структури менеджерів ресурсів;
- функціональності прикладних інтерфейсів менеджерів інформаційних ресурсів.

Запропоновані типові рішення формують концептуальну основу, а також регламентують складові практичного застосування методу. Ці рішення були використані у практичній розробці Web-застосування для навігації та доступу до інформаційних ресурсів в мережі Інтернет. Результати цієї розробки підтвердили практичну доцільність методу за рахунок спрощення процесів інтеграції, застосування єдиних механізмів взаємодії та прикладних інтерфейсів, зменшення витрат щодо розгортання компонентів та керування інформаційними ресурсами.

1. Дунаев С. Б. Доступ к базам данных и техника работы в сети. — М.: ДИАЛОГ-МИФИ, 1999. — 416 с.
2. Цимбал А.А., Аншина М.Л. Технологии создания распределенных систем. Для профессионалов. — СПб.: Питер, 2003. — 576 с.
3. Просиз Дж. Программирование для Microsoft .NET. — М.: Издательско-торговый дом "Русская Редакция", 2003. — 704 с.
4. Иванова Е. Б., Вершинин М. М. Java 2, Enterprise Edition. Технологии проектирования и разработки. — СПб.: БХВ-Петербург, 2003. — 1088 с.
5. Грищенко В.Н., Лаврищева Е.М. Компонентно-ориентированное программирование. Состояние, направления и перспективы развития // Проблемы программирования. —2002. — № 1-2. — С. 80-90.
6. Грищенко В.М. Систематизованый підхід до визначення програмних компонентів // Там же. — 2001. — № 3-4. — С. 23-30.

7. Грищенко В.Н. Формальные модели компонентного программирования // Там же. — 2003. — № 2. — С.42-57.
8. Эммерих В. Конструирование распределенных объектов. Методы и средства программирования интероперабельных объектов в архитектурах OMG/CORBA, Microsoft/COM и Java/RMI. — М.: Мир, 2002. — 510 с.
9. Грищенко В.М. Підхід до створення Web-застосувань на компонентній основі // Проблемы программирования. — 2004. — № 4. — С.15-25.
10. Секреты создания интрасетей. — СПб.: Питер, 1998. — 592 с.

Отримано 25.10.04

Про автора

Грищенко Володимир Миколайович,
канд. фіз.-мат. наук

Місце роботи автора:

Інститут програмних систем НАН
України, просп. Академіка Глушкова,
40,
Київ-187, 03187, Україна
Тел. (044) 252 4656