УДК 51:681.3.06

*Галич В. П., Васильєв А. О., Шевченко Р. С., Дорошенко А. Ю.*

# ЗАСТОСУВАННЯ ОБ'ЄКТНОЇ ТЕХНОЛОГІЇ В СИСТЕМАХ ОБРОБКИ КАРТОГРАФІЧНОЇ ІНФОРМАЦІЇ INTERNET/INTRANET

*Вихід комп 'ютерних застосувань в Інтернет і широке використання мережних технологій висуває нові вимоги до засобів програмування прикладних задач у розподілених мережних середовищах.. В статті описано особливості застосування об 'єктної технології для обробки картографічної інформації в геоінформаційтіх системах. Аналізуються можливості використання трирівневих архітектур та web-технологій в таких системах. Порівнюються два варіанти реалізації картографічних систем та наводяться переваги застосування об 'єктних та розподілених технологій.*

Застосування розподілених архітектур для створення сучасних інформаційних систем з багатьма користувачами вимагає вирішення принаймні двох серйозних проблем, які потребують матеріальних затрат. Перша проблема - це підвищення вимог до показників продуктивності апаратного забезпечення робочих станцій, а друга - забезпечення підтримки конфігурацій доступу до даних, що зберігаються в системі. Розв'язання цих проблем включає вирішення низки питань, пов'язаних зокрема з експлуатацією всіма користувачами певних спільних ресурсів, з територіальною віддаленістю клієнтів, можливою низькою якістю ліній зв'язку, перевантаженням мережі при збільшенні обсягу даних тощо. Традиційно ці питання можна вирішувати двома шляхами: екстенсивним та інтенсив-

ним. Перший з них полягає у збільшенні продуктивності апаратури і пропускної здатності мережі, прокладенні нових ліній зв'язку, перенесенні архівних даних з метою зменшення обсягу бази данігх. Зазвичай такий спосіб вимагає значних матеріальних затрат і часто у чистому вигляді є неприйнятним, особливо при великій кількості користувачів і високій швидкості збільшення обсягу бази даних. Інший шлях, інтенсивний, полягає у вдосконаленні організації обслуговування клієнтів та забезпеченні гнучкої архітектури її програмної підтримки насамперед за рахунок створення нових сервісів, спільних для користувачів інформаційної системи. Типовими прикладами для архітектур таких сервісів є CORBA [1-2] та DCOM [16]. Ці сервіси є сервісами проміжного рівня (middleware services), оскільки займають проміжне становище між даними і програмами, які їх обслуговують, з одного боку, та застосуваннями користувачів, що орієнтовані на конкретну предметну область,- з другого. Такі сервіси звичайно мають мінімальний інтерфейс користувача або навіть взагалі його не мають. Часто вони можуть бути реалізовані для багатьох різноманітних платформ.

Перед авторами цієї роботи стояло завдання розробки системи надання картографічних сервісів для представлення зображень планів міста користувачам у мережі Intranet (перевежно службовцям Головного управління архітектури та містобудування м. Києва), а також користувачам мережі Internet. Компанією GradSoft була розроблена типова архітектура систем, яка дає змогу отримувати векторні масштабовані плани міст, обирати типи об'єктів, які необхідно відобразити, та проводити пошук вулиці або будинку за адресою. Використання тривівневої архітектури системи та об'єктно-орієнтованих web-технологій дає змогу ефективно побудувати систему, яка б задовольняла ці вимоги.

## Географічні інформаційні системи (ГІС)

Побудова електронних карт, їх географічний аналіз та використання набуває все більшого поширення в інформаційних технологіях [3, 18]. Сучасні технології ГІС уже здатні виконувати не лише простий пошук та найпростіший аналіз при розв'язуванні проблем, що стоять перед організаціями та окремими користувачами, а й використовувати механізми узагальнення та повноцінного аналізу географічної інформації при прийнятті оптимальних рішень, що базуються на сучасних підходах та засобах візуалізації географічних даних. Згідно з визначенням [18] ГІС - це сучасна комп'ютерна технологія для картування та аналізу об'єктів і подій реального світу. Такі технології поєднують традиційні операції роботи з базами даних з перевагами візуалізації та географічного (просторового) аналізу, який є природним засобом обробки інформації, що може бути нанесена на карту. Ці особливості відрізняють ГІС від інших систем та забезпечують унікальні можливості для їх використання у вирішенні широкого спектру задач, пов'язаних з аналізом та прогнозом, виділенням головних факторів, причин та можливих наслідків, плануванням стратегічних та наслідків поточних рішень. Крім просторових запитів, проведення аналізу та обґрунтування рішень ГІС може виконувати також автоматичну побудову карт, яка є набагато простішою та гнучкішою, ніж у традиційних методах ручного або автоматизованого картографування. Процес починається з побудови картографічних баз даних, які можуть бути неперервними та не пов'язаними з масштабом. Далі, використовуючи таку базу даних, можливо створювати електронні карти (або їх тверді копії) будь-якої території, масштабу, з необхідним семантичним наповненням. Використання в ГІС сучасних технологій СУБД та Internet/Intranet дає можливість швидкого поновлення, експортування та розповсюдження географічних даних кінцевим користувачам. У цій статті описано підхід до реалізації таких систем на прикладі географічної інформації міста Києва.

Типова ГІС, зображена на рис. 1, складається з п'яти частин.

Апаратні засоби становить комп'ютерна платформа, на якій розгорнута ГІС, а програмне забезпечення містить функції та інструментарій, необхідні для зберігання, аналізу та візуалізації географічної інформації. Дані - найважливіший компонент ГІС. Це інформація про просторове положення об'єктів, а також пов'язана з ними семантична інформація. Виконувачі - це персонал розробників, що працюють з програмними продуктами і розробляють їх застосування для розв'язування конкретних задач. Множину методів утворюють обрані плани та правила роботи кожної ГІС, що складаються відповідно до специфіки завдань кожної організації.

Далі в роботі використовуються такі поняття картографічних даних, як масштаб, географічний об'єкт, шар, растр, вектор та генералізація [4].

*Масштабом* називають відношення довжини нескінченно малого відрізка на геозображенні до довжини відповідного нескінченно малого відрізка на поверхні еліпсоїда або кулі. Існує велике розмаїття інформації, яка повинна бути нанесена на карту. Будь-яку одиницю такої інформації прийнято називати *географічним об'єктом*. Для

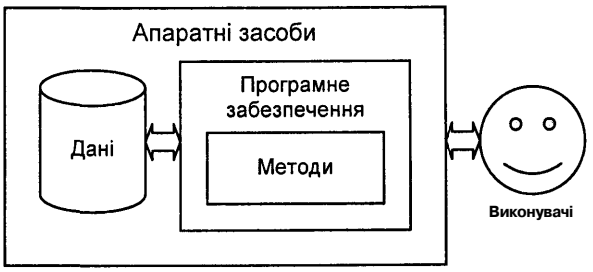


**Рис. 1. Структура ГІС**

побудови карт перш за все класифікується вся множина об'єктів карти. Першим кроком класифікації є виділення *шарів* картографічної інформації, де кожен шар містить певну множину об'єктів. Для карти міста Києва такими шарами можуть бути берегова лінія Дніпра та водосховищ, парки та зелені зони, система вулиць та кварталів. Далі множину шарів теж можна класифікувати, наприклад на шари, що є ландшафтними особливостями, шари забудов тощо. Існує відповідність між класифікацією шарів із семантичної точки зору та їх відображеннями на картографічних матеріалах певного масштабу [4]. Вона описана у відповідних державних, галузевих стандартах та керівних документах.

Графічне відображення цієї інформації звичайно може бути оформлене у двох форматах: *растровому* або *векторному*. Растрове зображення визначає матрицю елементарних пікселів, які формують зображення. Проте семантичний аналіз растрового зображення, наприклад для виділення об'єктів, визначення масштабу або пошуку зразка, є досить складною задачею, що може вирішуватися за допомогою систем штучного інтелекту, і тому не придатний для використання у критичних до часу елементах інтерактивної системи. З іншого боку, кінцевим елементом інтерфейсу користувача є саме растрове зображення, яке формується за допомогою пікселів дисплея. Векторне зображення - це математичний опис зображення як набору геометричних примітивів. Використання векторного формату дає змогу досить ефективно реалізувати семантичні операції над картографічними об'єктами.

*Генералізацією* називають узагальнення зображень малих масштабів відносно більших, яке здійснюється у зв'язку з тематичним призначенням або технічними умовами отримання самого зображення [4].

## Функціональні та технологічні вимоги до системи

Функціональні вимоги до картографічного застосування можна розподілити на вимоги до сервера застосувань, сервера баз даних та клієнтської частини. До функцій сервера застосувань можна віднести: навігацію користувача по масштабованому плану міста, вибір району для подальшого його детального відображення, зміну масштабів детального відображення вибраного району, обробку запитів на пошук адреси та відображення знайденої інформації. Функціями сервера баз даних є: зберігання всіх необхідних даних та виконання запитів, які надходять від сервера застосувань. До клієнтської частини ставляться вимоги відображення інформації, отриманої від сервера застосувань, та формування і передача запитів до нього.

Звичайно картографічне застосування повинне задовольняти такі технологічні вимоги, як масштабованість, відкритість, переносність, ізольована розробка та «легкість» клієнтів, що означає зосередження всієї обробної частини застосування на серверній стороні. *Масштабованість* - це можливість підвищення обчислювальної потужності комп'ютерної системи (наприклад, збільшення кількості операцій або тракзакцій за одиницю часу) за рахунок збільшення кількості обчислювальних модулів або заміни їх на потужніші. *Відкритість* гарантує, що система безпроблемно інтегрується в існуючі або нові застосування. Крім того, обчислювальна система повинна функціонувати в гетерогенних і, що найбільш важливо, розподілених середовищах. Відкритість безпосередньо пов'язана із поняттям масштабованості. Вона його розширює, оскільки вимагає одночасної підтримки багатьох платформ, мережних середовищ та серверів баз даних. Крім того, застосування повинне забезпечувати легке підключення зовнішніх застосувань. Практично це означає, що воно повинне мати відкритий інтерфейс користувача API та підтримку технологій COM (DCOM) або CORBA, а також підтримку існуючих у даній галузі стандартів [5-6]. Вимога *переносності* забезпечує виконання застосування коду на інших платформах без істотної втрати функціональності. Фактично вона є окремою частиною вимоги відкритості.

*Ізольована розробка* - це властивість розподілених застосувань через свою модульну основу дозволяти ізольовані одне від одного створення і заміну модулів (компонент). Вся система розбивається на автономні модулі, робота над якими може проводитись окремо від інших [16]. В той же час модулі можуть взаємодіяти між собою. Для цього вони повинні підтримувати протоколи і інтерфейси, що визначають спільні принципи їх взаємодії. Оскільки методи, що існують у модулях, ізольовані від методів інших модулів, вони можуть розроблятися незалежно. Таким чином, міра реалізації компонент не залежить від стану коду в інших частинах системи. Стає можливою паралельна робота декількох команд над різними частинами застосування або системи. Взаємодія між різними модулями відбувається через встановлені протоколи та інтерфейси.

*Легкі клієнти* у розподіленій системі надають можливість перенести всю функціональну логіку інформаційної системи на її серверну частину. У цьому випадку клієнти, з якими спілкується користувач, можуть бути виконані невеликими і легкими. Системні ресурси користувача виявляються вільнішими, а весь тягар функціональної логіки реалізується високопродуктивним сервером (або мережею серверів). При цьому клієнт може мати доступ до практично необмеженої кількості сховищ інформації та інших об'єктів. З'являється можливість створення «легких» компонент, придатних для швидкого завантаження через мережу Internet і запуску на комп'ютері клієнта. До такого роду застосувань можна віднести аплети Java та ActiveX компоненти.

### Архітектура картографічних застосувань

У більшості випадків структура вхідної задачі загалом визначає архітектуру системи. Картографічні застосування, по-перше, повинні мати підсистему зберігання векторних даних, інтегровану з системою вибірки. По-друге, має існувати система обробки вибраних даних, яка взаємодіє з іншими підсистемами в термінології картографічних понять: масштаб, шар, географічний об'єкт. По-третє, має бути система відображення картографічної інформації та взаємодії з користувачем. Підсистема відображення повинна реалізувати відображення векторних даних у растровій формі, яка відповідає відображенню цих даних на графічних дисплеях. Підсистема зворотного зв'язку має організовувати кінцевий інтерфейс користувача. І останнє: якщо мова йде про систему з інтерфейсом всесвітньої мережі WWW (далі скорочено - веб), то повинна існувати також підсистема доставки та кешування елементів кінцевого інтерфейсу. Таким чином, типова архітектура подібних систем має відповідати схемі, наведеній на рис. 2.

### Реалізації систем картографічних застосувань

У даній роботі були побудовані дві системи, що реалізують вищенаведену архітектуру картографічних застосувань. Перша була втілена на основі Java з використанням таких засобів: веб-браузер Internet Explorer 5, JDBC, веб-сервер Apache [19], modCBroker [8], бібліотека обробки графіки Acme (www.acme.com). Картографічна інформація містилася в базі даних Oracle у вигляді векторних об'єктів: ліній, полігонів, прямокутників, кривих, їх дані відображалися засобами Java та JDBC. Оскільки відображати картографічні дані потрібно було у графічному форматі, необхідно було здійснювати перетворення вибраних векторних об'єктів у растрові малюнки. Це було здійснено мовою Java із застосуванням спеціалізованої бібліотеки Acme. Остання дає змогу ефективно продукувати такі широковживані растрові формати, як jpeg, gif та ppm. Функціональні засоби було реалізовано в сервлетах із використанням модуля Apache ModCBroker, розробленого компанією GradSoft. Даний модуль здійснює прозору трансляцію HTTP запитів до сервера CORBA. Згенеровані сервлетами зображення посилаються в гіпертекстовий потік для браузера клієнта. Користувач має справу із малюнком карти Києва та навігаційними кнопками, які дають змогу переміщатися по карті та збільшувати або зменшувати масштаб. На рис. 3 зображено технологічні елементи даної реалізації.

Поєднання технологій Java та CORBA має свої переваги та недоліки. До переваг можна віднести здатність Java спрощувати розповсюдження коду у великих системах CORBA, оскільки такий код

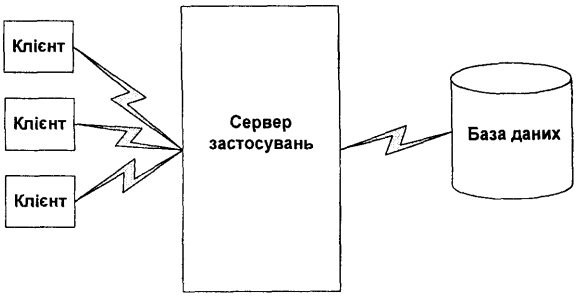


**Рис. 2. Архітектура застосування**

може застосовуватися і керуватися централізовано із сервера [17]. У разі потреби слід лиш поновити код на сервері і потім надати клієнтам можливість його отримувати. До позитивів можна зарахувати також той факт, що Java є зручним засобом написання об'єктів CORBA [17]. Вмонтовані можливості Java для мультипоточності, збирання сміття та обробки помилок полегшують написання мережних об'єктів. Об'єктна модель Java доповнює модель CORBA, і вони обидві використовують концепцію інтерфейсів для відмежування визначення об'єкта від його реалізації. До недоліків можна зарахувати низьку продуктивність Java на таких складних математичних обчисленнях, як обробка зображень та криптографія.

Інший варіант реалізації було одержано в системі GradMap, що була реалізована компанією GradSoft. Вона має три рівні: рівень бази даних, рівень застосувань та рівень клієнта. На рівні бази даних працює сервер баз даних Oracle, який зберігає геометричні образи об'єктів, а також пов'язану семантичну інформацію: назва, описи тощо. Як сервери застосувань використовуються об'єкти CORBA. Завданням кожного окремого сервера є аналіз запиту користувача, формування відповідної об'єктної моделі карти, растеризація, генералізація, кешування, формування геометричного відображення карти та подання відповідних результатів клієнту. Клієнтом системи є веб-браузер, який забезпечує користувачеві всі доступні функції системи, формує запит і представляє результат роботи системи.

Функції взаємодії сервера застосувань та браузера для клієнта реалізуються засобами ModCBroker [8], а взаємодія сервера застосувань та СУБД

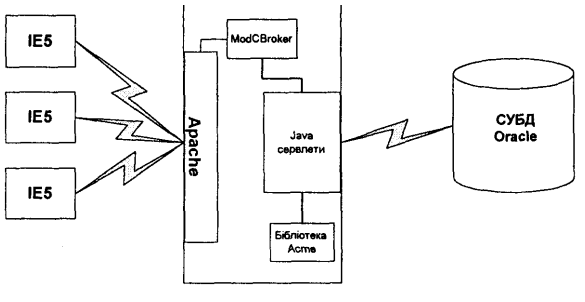


**Рис. 3. Технологічні елементи реалізації на основі Java**

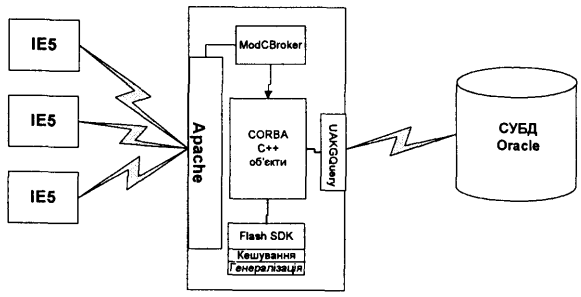


**Рис. 4. Технологічні елементи системи GradMap**

виконується на основі технології UAKGQuery - обидва продукти розроблені компанією GradSoft [8, 9]. Основна їх перевага полягає в тому, що вони дають змогу виконувати запити зчитування, запису даних об'єктами CORBA до будь-якої системи управління БД.

Технологічні елементи реалізації даної системи зображено на рис. 4.

Побудована об'єктна модель даних завжди потребує деякого графічного представлення. Для запису графічної інформації в системі була використана бібліотека FlashSDK (www.macromedia.com). Вона дає змогу сформувати графічні об'єкти та записати їх у файл формату .swf, що дозволяє використовувати переваги Flash технології. На рис. 5 наведено приклад зображення засобами універсальної мови моделювання UML спрощеної схеми класів сервера застосувань. На рівень прикладних об'єктів тут винесені такі поняття, як шар (Layer), що є сукупністю об'єктів зображення (ImageObject), таких як полігон або незамкнутий контур. Представлення даних, як і параметри конкретного запиту користувача, ізольовано від шару цих об'єктів за допомогою спеціального класу-обробника (DataHandler). Інші елементи, такі як блок генералізації або блок кешу запитів, на цій схемі не представлені.

Розробка GradMap показала, що при сучасному розвитку технології моделювання UML модель системи може грати лише допоміжну ілюстративну роль у великих проектах. Побудована повна UML модель системи містить понад 300 об'єктів, і тому використовувати програмні засоби роботи з моделлю виявилось практично неможливо внаслідок їх низької продуктивності.

На рис. 6 наведено елемент інтерфейсу кінцевого користувача картографічного застосування у системі GradMap.

## Висновки

Досвід розробки розподіленої системи картографічних застосувань виявив певні особливості таких розподілених комплексів. Перша особливість - це використання типової архітектури з можливістю вибору елементів реалізації. Наприклад, у разі потреби можна використовувати CORBA API підсистеми географічних об'єктів (приклад GradMap) у комбінації з серверним генеруванням растрового

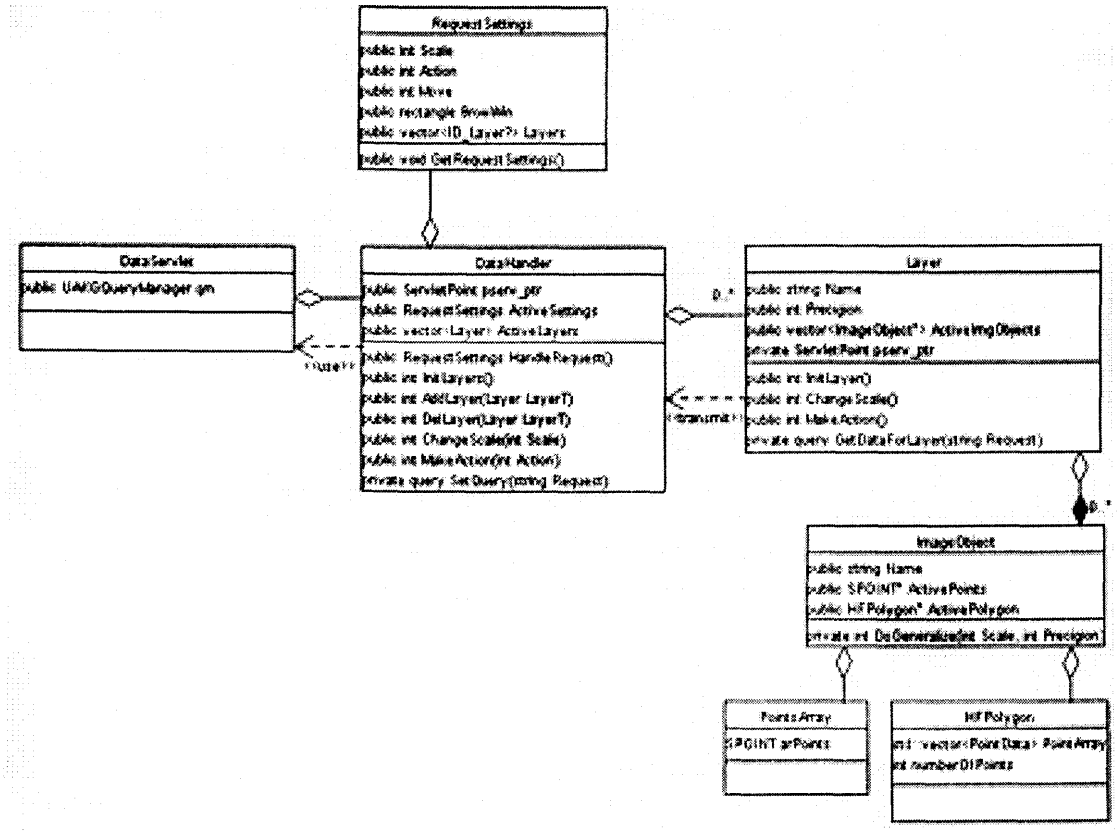


**Рис. 5. Спрощена діаграма UML для класів сервера бізнес-логіки**

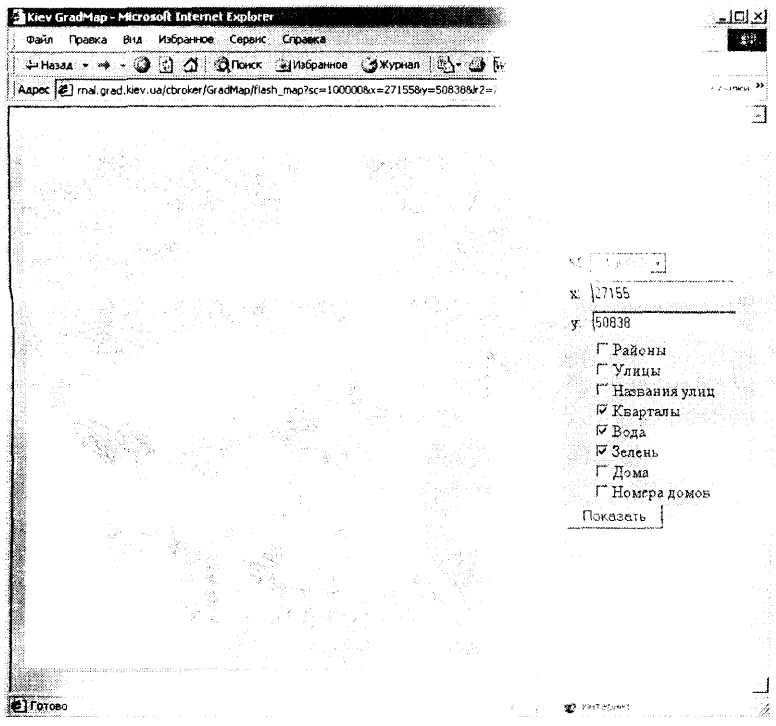


**Рис. 6. Приклад інтерфейсу користувача**

зображення засобами Java (перший варіант реалізації). Друга характерна риса - це використання багаторівневої архітектури, де кожен рівень відповідає певному рівню абстракції, тобто бази даних, геооб'єктів, інтерфейсу користувача, взаємодії з веб. І третя особливість - це використання компонентної моделі. В середовищі CORBA об'єкт стає інтелектуальною програмною одиницею, незалежного компонентою, завершеною частиною програмного коду із такими особливостями:

- компонента сама по собі не є застосуванням;
- компонента не залежить від адресного простору, мережного протоколу, алгоритмічних мов, операційних систем, засобів програмування;
- компонента має добре розвинений інтерфейс.

Компонента виступає одиницею роботи і розподіленості в розподілених об'єктних системах. Інфраструктура розподілених об'єктів CORBA забезпечує прості механізми для того, щоб компоненти стали автономними, самокерованими і взаємодіючими. Технологія розподілених об'єктів CORBA дає змогу збирати в єдине ціле складні інформаційні клієнт-серверні системи, просто зв'язуючи І розширюючи їх компоненти, причому можна модифікувати об'єкти, не впливаючи на інші компоненти або на спосіб їх взаємодії. Це потужний засіб, який успішно вирішує проблему, для якої він створювався, а саме проблему взаємодії. Уже сьогодні намічаються тенденції зсуву архітектури складних обчислювальних систем від однорівневої до дво- та багаторівневої архітектури [10]. Розподілені об'єкти та Web дають змогу розбити важкі монолітні застосування на більш керовані легкі компоненти, які мають властивість існувати в гетерогенних середовищах.

У майбутньому досвід використання багаторівневих архітектур буде поширюватись на більш спеціалізовані системи, такі як геоінформаційні. Саме застосування таких архітектур та веб-технологій дає змогу розширити можливості систем обробки картографічної інформації. Застосування об'єктних технологій дає змогу зменшити загальну вартість складних систем за рахунок повторного використання коду і можливості ретельної доробки компонент на різних стадіях життєвого циклу програми.

1. *Object Management Group, formal/98-12-0]:* The Common Object Request Broker: Architecture & Specifications. CORBA/IIOP 2.3.1, 712 p. (ftp://ftp.omg.org/pub/formal/98-12-01.pdf).
2. *Object Management Group. Oma:* A Discussion of the Object Management Architecture January 1997.- 44 p. (www.omg.org/library/oma/oma-all.pdf).
3. *Open Gis Consortium.* Open CIS Abstract Specification.- 99-AS-RFP009 24 June 1999 (http://www.opengis.org/).
4. Геоинформатика.Толковый словарь основных терминов // Баранов Ю. Б., Берлянт А. М., Капралов Е. Г., Кошкарев А. В., Серапинас Б. Б., Филиппов Ю. А. / Под редакцией Берлянта А. М. и Кошкарева А. В.- М.: ГИС-Ассоциация- 204 с. (http: // 193.233.5.209/intbl/gis_serv/ourj3ubl/Dictionary/index_slov.htm).
5. ГОСТР 50828-95 Геоинформационное картографирование. Пространственные данные, цифровые и электронные карты. Общие требования.
6. ГОСТР 51353-99 Геоинформационное картографирование. Метаданные электронных карт. Состав и содержание.
7. *Fowler.* Λ/Analysis Patterns and Business Objects- OOPSLA'96, 1996.
8. *ShevchenkoR, KrisanovS.* Mod_cbroker Programming Guide, 2001.
9. *Shevchenko R.* UAKGQuery: Programming Guide, 2001.
*IQ.Shevchenko R., Doroslienko A.* A Method of Mediators for Building Web Interfaces of CORBA Distributed Enterprise Applications. Lecture Notes in Informatics // V. 4. Proceeding of Information Systems Technology and its Applications, 2001; Gesselschaft für Informatik, 2001.- ISBN 3-88579-331-8.

11. *Shevchenko R., Dowshenko A,* Techniques for Increasing Performance of CORBA Based Distributed Applications. Parallel ComputerTeclinologies, Proc. 6-th int. conf. PACT2001 LNCS-V. 2127.-P. 319-328.- Springer-Verlag, 2001.
12. *ВебердЖ.* Технология Java. -СПб.: BHV, 1997.
13. *Stroustrup B.* The C++ Programming Language (3rd edition). Addison-Wesley Longman, Reading, MA- 1997- ISBN 0-201-88954.- 4.- 920 p. Softcover.
14. *Stroustrup B.* The Design and Evolution of C++. Addison-Wesley, Reading, MA.- 1994.- ISBN 0-201-54330-3.-472 p. Softcover. (Often called D&E).
15. Открытые системы: концепция или реальность // Открытые системы- 1993- № 4.
16. *Семихатов С.* Технологии WWW, Corba и Java в построении распределенных объектных систем, http: // www.javable.com/docs/jav_dist/.
17. *Орфали Р., ХаркиД.* Java и CORBA в приложениях клиент-сервер. Лори, 2000-ISBN 5-85582-092-0.
18. Географические информационные системы-М.: ООО Дата+. http://www.dataplus.ru/.
19. *Ben Laurie and Peter Laurie.* Apache. The Definitive Guide.-O'Reilly.- 1997.-ISBN 1-56592-250-6.

*V. P. Galych, A. O. Vasyliev, R. S. Shevchenko, A. Y. Doroshenko*

## APPLICATION OF OBJECT TECHNOLOGY IN INTERNET/INTRANET CARTOGRAPHIC INFORMATION PROCESSING SYSTEM

*Widespread of Internet computer applications and network technologies puts forward the new requirements to ways of programming applied tasks in the distributed network environments. Features of object technology deployment for processing the cartographic information in geoinformation systems are described. Opportunities for application of three-level architecture and Web technologies in such systems are analyzed. Two variants of realization of cartographic systems are compared and the advantages of application object and distributed technologies are resulted.*