

Побудовано математичні моделі високого ступеня адекватності для процесів формування максимальної та мінімальної цін американського долара відносно канадського долара. Запропоновано метод побудови регресійних моделей з використанням схеми адаптивного прогнозування та індикаторів технічного аналізу. Для прогнозування напрямів руху ціни використано моделі, які прийнято відносити до інтелектуального аналізу даних: логістична регресія та дерево класифікації. Найкращими моделями виявилися ті, що являють собою симбіоз регресійної моделі та логістичної регресії. Зокрема, ймовірність коректного прогнозування руху мінімальної ціни становила 75,39%, що є задовільним результатом для процесів цього класу.

Упродовж подальших досліджень доцільно розглянути індикатори інших типів, а також застосувати удосконалені схеми адаптивного прогнозування з можливістю покращення якості попередньої обробки вхідних даних, що використовуються для побудови моделей.

1. Лукашин Ю.П. *Адаптивные методы краткосрочного прогнозирования*. – М.: Финансы и статистика, 2003. – 414 с. 2. Зельнер А. *Байесовские методы в эконометрии*. – М.: Статистика, 1980. – 438 с. 3. Chatfield C. *Time series forecasting*. – London: Chapman & Hall, 2000. – 267 p. 4. Згуровский М.З., Подладчиков В.Н. *Аналитические методы калмановской фильтрации*. – К.: Наукова думка, 1995. – 285 с. 5. Бідюк П.І. *Системний підхід до прогнозування на основі моделей часових рядів // Системні дослідження та інформаційні технології*. – 2003. – № 3. – С. 88–110. 6. Zgurovsky M.Z., Bidiuk P.I., Terentyev O.M. *Method of constructing Bayesian networks based on scoring functions // Cybernetics and System Analysis*, 2008, Vol. 44, No.2, pp. 219-224. 7. <http://www.mataf.net/en/tools/home>. 8. Nong Y. *The Handbook of Data Mining*. – New Jersey: Arizona State University Publishers, 2003. – 1201 p. 9. Altman E.I., Avery R.B., Eisenbeis R.A., Sinkey J. *Application of Classification Techniques in Business, Banking and Finance*. – Greenwich: JAI Press, 1981. – 418 p. 10. Hosmer D.W., Lemeshow S. *Applied Logistic Regression*. – New York: John Wiley & Sons, Inc., 2000. – 380 p.

УДК 681.3, 621.3

Ю. Морозов, І. Пастернак

Національний університет “Львівська політехніка”,  
кафедра електронних обчислювальних машин

## МОДЕЛЬ ОБ’ЄКТНОЇ КЛІЄНТ-СЕРВЕРНОЇ ВЗАЄМОДІЇ

© Морозов Ю., Пастернак І., 2011

**Запропоновано варіант моделі об’єктної взаємодії між клієнтом та сервером. Наведено мережну клієнт-серверну взаємодію, а також запропоновано компонентний підхід, який полегшує розроблення мережних інтерфейсів.**

**Ключові слова:** компонентний підхід, моделі об’єктної взаємодії, клієнт-серверна взаємодія, мережний інтерфейс.

**The variant the object model of interaction between client and server. The client-server interaction in network. Also proposed component approach, which facilitates the development of network interfaces.**

**Keywords:** component approach, models object interaction, client-server interaction, network interface.

### Вступ

Програмування як рід людської діяльності з’явився порівняно недавно, але стрімкий розвиток комп’ютерних технологій дозволяє нам сказати, що комп’ютеризація займає значне місце в сучасному світі. В історії комп’ютерних технологій, що пов’язані зі збиранням, обробкою

інформації та розв'язанням задач виділяють такі етапи: поява комп'ютерів, що зберігали програму в пам'яті, створення мов програмування високого рівня та створення компонентного програмування. Загальні методи об'єктно-орієнтованого аналізу та проектування програмного забезпечення значно вплинули на більш спеціалізовану область комп'ютерної інженерії – компонентний підхід до створення програмного забезпечення. Компонентний підхід є частковим випадком об'єктної орієнтованості і він вирішує проблему інкапсуляції складності.

Компонентні об'єктні середовища вважаються найсучаснішим та природним фундаментом для накопичення та використання знань з програмування. Таке середовище ґрунтується на компонентній об'єктній моделі. Воно містить готові компоненти, а також інструментальні засоби, які дають змогу вибрати компоненти, налагодити їх, пов'язати між собою та об'єднати.

### Огляд літературних джерел

Компонентні середовища володіють всіма перевагами об'єктно-орієнтованого підходу [4–6]:

1. Інкапсуляція об'єктних компонентів приховує складність реалізації, роблячи доступним тільки представлений зовні інтерфейс.

2. Наслідування дає змогу розвивати створені раніше компоненти, при цьому не порушуючи цілісності об'єктної оболонки.

3. Поліморфізм дає можливість групувати об'єкти характеристики, які з деякого погляду схожі.

Найчастіше розподілені об'єкти (компоненти) працюють у конфігурації клієнт-сервер. Самі сервери є об'єктами, які реагують на запити і надають клієнту сервіси або ресурси.

### Постановка задачі

Розробити модель мережної взаємодії, яку можна було б компонувати на рівні інтерфейсу.

### Основні результати досліджень

Модель об'єктної взаємодії – це найшвидший спосіб зв'язку для передавання станів об'єктів (рис. 1). Цей протокол використовує безпосередню передачу стану віддаленого об'єкта за значенням [1–3].

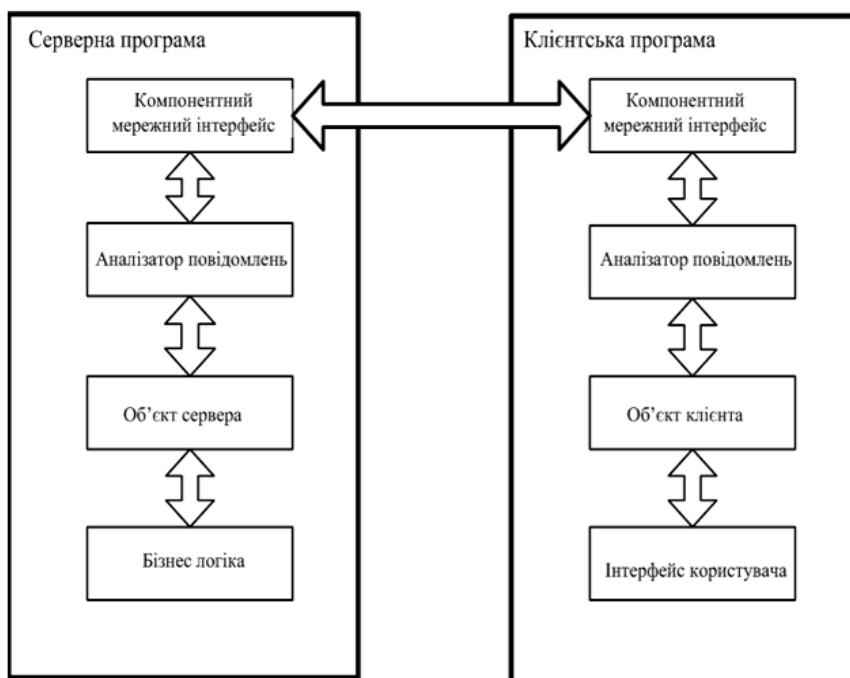


Рис. 1. Модель об'єктної взаємодії

Модель об'єктної взаємодії використовує протокол HTTP для передавання стану віддаленого об'єкта, відформатованого у бінарному форматі каналом зв'язку. Під час реалізації модель об'єктної взаємодії використовує серіалізацію та десеріалізацію, які відбуваються як на стороні клієнта, так і на стороні сервера. Клієнт отримує можливість відправляти прості і складні об'єкти на сервер. Ці значення клієнтських об'єктів можуть бути прив'язані до об'єкта сервера шляхом додавання метаданих, тобто тегів, до класу об'єкта [7]. Налаштування моделі об'єктної взаємодії комунікацій трохи складніше, ніж створення зв'язку клієнта з web-сервісом. Модель об'єктної взаємодії у використанні показала себе ефективнішою ніж інші, тому що навантаження каналу зв'язку даними в двійковому форматі менше в результаті меншого розміру пакета.

Локальним об'єктом є об'єкт за значенням (ОПЗ). Код методів такого об'єкта виконується локально. Якщо ОПЗ було отримано з віддаленого боку, то необхідний код повинен або бути наперед відомий обом сторонам, або бути динамічно завантажений. Щоб це було можливо, запис, що визначає ОПЗ, містить спеціальне поле зі списком адрес, звідки може бути завантажений код [8]. У ОПЗ можуть також бути і віддалені методи. У ОПЗ можуть бути поля, які передаються разом з самим ОПЗ. Вони також можуть бути ОПЗ, формуючи таким чином списки, дерева або довільні граfi. ОПЗ можуть мати ієрархію класів, включаючи множинне наслідування і абстрактні класи.

Мережні інтерфейси можуть взаємодіяти один з одним як на одній локальній машині, так і по мережі. Вони можуть спілкуватися один з одним, незалежно від їх конкретної реалізації, "прописки" в розподіленій системі, платформи і мови їх реалізації. Вони утворюють нижній шар архітектури проміжного шару, що забезпечує технологічну платформу інтероперабельності. Семантику об'єктів на цьому рівні не беруть до уваги. Завдання мережних інтерфейсів — інтегрувати розподілені системи, дати можливість програмам, що написані різними мовами та працюють у різних вузлах мережі, взаємодіяти одна з однією так само просто, наче вони знаходяться в адресному просторі одного процесу.

Компонентна модель взаємодії (КМВ) описує стандартний каркас мережного інтерфейсу. КМВ побудована на основі Enterprise Java beans (EJB) і фактично є його незалежним від мови розширенням. КМВ надає абстракцію суті, яка може надавати і отримувати сервіси через певні іменовані інтерфейси та порти. Модель КМВ надає контейнер компонентів, в якому можуть поставлятися програмні компоненти. Контейнер надає список служб, які може використовувати компонент. Це служби повідомлень, авторизації, сховища і управління транзакціями. Ці служби найчастіше використовуються розподіленими додатками. Перенесенням реалізації цих сервісів від необхідності реалізації самим додатком у функціональність контейнера додатку, можна значно понизити складність реалізації власне компонентів.

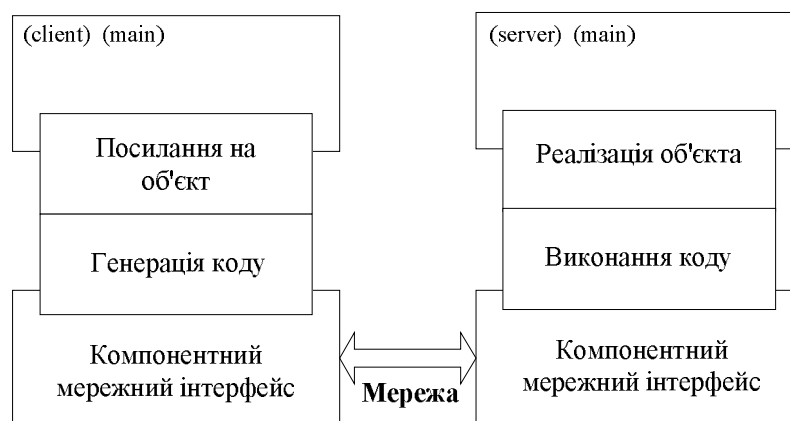


Рис. 2. Інфраструктура мережної взаємодії з використанням компонентного мережного інтерфейсу

Об'єктна клієнт-серверна взаємодія складається із серверної та клієнтської частин. До неї можна застосувати рівневий підхід. Кожен рівень клієнтської частини обслуговує відповідним

рівнем серверної частини за взаємодії через мережу: посилання на об'єкт – реалізація об'єкта, генерація коду – його виконання. Ці частини містять компонентний мережний інтерфейс, через який здійснюється взаємодія між клієнтом і сервером у мережі. У разі взаємодії цих частин у мережі, за посиланням на об'єкт в клієнтській частині завантажується реалізація об'єкта із серверної частини [9, 10]. Як правило, на стороні сервера знаходиться портативний адаптер об'єкта, який скеровує запити на інші сервери для балансування навантаження на мережу. Завдяки такій структурі спільна робота мережного обладнання й програмного забезпечення стає набагато простішою й зрозумілішою.

### Висновки

Проаналізовано компонентний підхід до клієнт-серверної взаємодії. Наведений підхід полегшує розроблення мережних інтерфейсів. Також запропоновано модель компонентної об'єктної взаємодії між клієнтом і сервером.

1. Булах Е. В. Средства доступа к базам данных в Internet и свободно доступная СУБД POSTGRES95. Центр информационных технологий. 2. Веселов В. А., Долженков. XML и технологии баз данных 16.05.2000. 3. Галактионов В. В. Java-технология в распределенных системах с CORBA-архитектурой. 6. Гербер Берри. Microsoft Exchange Server 5.5 для профессионалов. — СПб: Питер Ком, 1999. — 528 с. 4. Дунаев С. INTRANET-технологии. WebDBC. CGI. CORBA 2.0. Netscape Suite. Borland IntraBuilder. Java. JavaScript. LiveWire/ — М.: Диалог-МИФИ, 1997 — 288 с. 5. Семихатов С. Технологии построения распределенных объектных систем. 10. Федоров А. Технологии для Web-сервисов // КомпьютерПресс. — 2002. — N 6. — С. 32—36. 6. Федоров А. Платформы и средства создания Web-сервисов // КомпьютерПресс. — 2002. — N 6. — С. 38—53. 7. Эммерих В. Конструирование распределенных объектов. Методы и средства программирования интероперабельных объектов в архитектурах OMG/CORBA, Microsoft/COM и Java/RMI. Пер. с англ. — М.: Мир, 2002. — 510 с. 8. ANSI/NISO Z39.50-1995. Information Retrieval (Z39.50): Application Service Definition and Protocol Specification. Z39.50 Maintenance Agency Official Text for Z39.50-1995, July 1995. 10. Dave Raggett, Arnaud Le Hors, Ian Jacobs, editors. HTML 4.01 Specification. World Wide Web Consortium, 1999.