

ляет обеспечить гарантированное качество связи при решении задач сетевого уровня по показателям – требуемая полоса пропускания или требуемое время доведения.

#### Литература

1. Вегешна Ш. Качество обслуживания в сетях IP. – М.: Издательский дом «Вильямс», 2003. – 368 с.
2. Остерлох Х. Маршрутизация в IP-сетях. Принципы, протоколы, настройка. – С.Пб.: BHV, 2002. – 512 с.
3. Олифер В., Олифер Н. Искусство оптимизации трафика // LAN. 2001. №12. – С. 38-47.

*Разрабатывается объектно-реляционная алгебра, предназначенная для сопряжения объектно-ориентированной ГИС и РБД. ОРА алгебра представляет собой набор операторов, использующих коллекции классов в качестве аргументов, и возвращающие коллекции классов в качестве результата. Определены двадцать один оператор, которые объединены в три группы: теоретико-множественные операторы, специальные ОРА операторы и особые географические операторы. Использование возможностей, предоставляемой ОРА, позволит сопрягать ГИС и РБД более эффективно, чем в технологии ArcSDE. Программный адаптер, реализующий сопряжение на основе ОРА, позволяет повысить скорость разработки ГИС.*

#### Введение

Поддержание в актуальном состоянии информационной среды по геопространственным природно-техническим комплексам является важной задачей в народном хозяйстве. Для управления геоданными используются геоинформационные системы (ГИС). Современные ГИС строятся по объектно-ориентированному принципу, а свои данные хранят в реляционных базах данных (РБД). Возникает потребность унифицировать технологии сопряжения ГИС с РБД и за счет этого сократить затраты на разработку. Современные технологии сопряжения ГИС и РБД не решают вопрос стыковки объектной и реляционной модели. Такие технологии как ArcSDE только расширяют возможности пространственного анализа геоданных. В статье предлагается расширить возможности сопряжения, включив в него объектно-реляционное преобразование. В статье описана разработанная на основе реляционной алгебры объектно-реляционная алгебра (ОРА), которая позволяет работать с классами, а не с реляционными отношениями. Алгебра должна дать возможность работать со свойствами класса, классами и коллекциями классов как с элементами множества, кортежами и мно-

4. Segall A. The modeling of adaptive routing in data-communications networks. // IEEE Trans. on communications, 1975. Vol. 25. №1. – P. 85-95.

5. Лемешко А.В., Беленков А.Г. Динамическая модель комплексного решения задач маршрутизации и абонентского доступа в территориально-распределенных телекоммуникационных сетях // Открытые информационные и компьютерные интегрированные технологии: Сб. науч. тр. Нац. аэрокосмич. ун-та им. Н.Е. Жуковского „ХАИ“. – Харьков: НАКУ. 2003. – Вып. 18. – С. 134-139.

УДК 004.9:004.64

## АЛГЕБРА СОПРЯЖЕНИЯ ГЕОИНФОРМАЦИОННЫХ СИСТЕМ

**С.В. Ляховец**

*Харьковский технический университет радиозлектроники просп. Ленина, 14, Харьков, Украина, 61166  
E-mail: itl1155@online.kharkov.ua.  
Контактный тел.: 702-14-46.*

**Г.Г. Четвериков**

*Кандидат технических наук, доцент кафедры ПО-ЭВМ Харьковского технического университета радиозлектроники просп. Ленина, 14, Харьков, Украина, 61166  
Контактный тел.: 702-14-46.*

жествами соответственно. В ОРА введены также необходимые операции для географических манипуляций с объектами.

#### Разработка общих принципов ОРА

Обозначим латинской большой буквой класс, например  $A$ . При необходимости к букве будем добавлять номер. Обозначим коллекцию классов, как  $[A]$ . Атрибуты (свойства) класса будем обозначать большими латинскими буквами. Например,  $X, Y, \dots, Z \in A$ . При необходимости к букве будем добавлять номер.

ОРА представляет собой набор классов-коллекций и класса операций  $CO$ , использующего свои методы, с некоторыми аргументами – коллекциями классов, и возвращающие коллекцию классов в качестве результата. Для специальных географических манипуляций существует класс  $COG$ , аналогичный  $CO$ . Методы классов  $CO$  и  $COG$  будем называть операциями. Таким образом, метод класса операций  $CO$  выглядит как метод  $f$  с коллекциями классов в качестве аргументов:

$$[R] = CO.f([R1], [R2], \dots, [Rn])$$

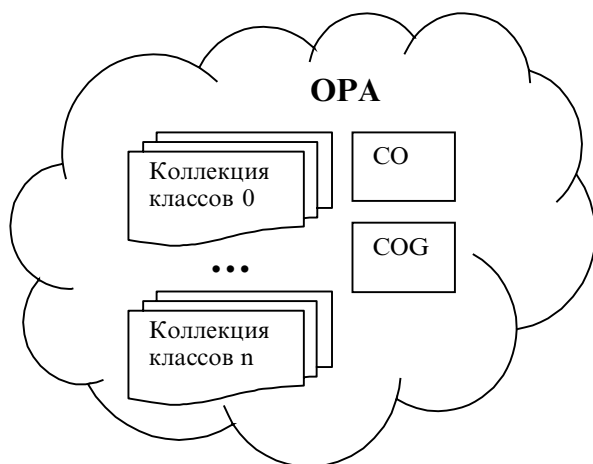


Рисунок 1. Состав объектно-реляционной алгебры

ORA является замкнутой, т.к. в качестве аргументов в операции  $CO$  можно подставлять другие классы, подходящие по типу:

$$[R] = CO.f(CO.f1([R11], [R12], \dots), CO.f2([R21], [R22], \dots), \dots)$$

Таким образом, в выражениях ORA можно использовать вложенные выражения сколь угодно сложной структуры.

Каждый класс и коллекция классов обязаны иметь уникальное имя в пределах базы данных и информационной системы.

Определим восемь операций класса  $CO$ , основываясь на реляционной алгебре, и тринадцать операций класса  $COG$ , которые применяются в технологии ArcSDE. Все эти операции можно определить в три группы. Теоретико-множественные операции: Объединение, Пересечение, Вычитание, Декартово произведение. Специальные операции: Выборка, Проекция, Соединение, Деление. Особые географические операции: Равенство, Касание, Внутри, Содержит, Скрещивание, Частичное совпадение, Нет пересечения, Наложение, Пересечение, Разность, Симметрическая разница, Объединение, Функция отсечения. Не все они являются независимыми, т.е. некоторые из этих операций могут быть выражены через другие операции. Некоторые операции требуют, чтобы классы и, соответственно, коллекции классов имели одинаковые типы.

**Определение 1.** Будем называть классы *совместимыми по типу*, если они имеют идентичные типы, а именно, классы имеют *один и тот же набор атрибутов*. Будем называть коллекции классов совместимыми по типу, если типы классов в коллекциях совпадают.

Некоторые классы не являются совместимыми по типу, но становятся таковыми после некоторого переименования атрибутов. Для того, чтобы такие классы можно было использовать в  $CO$ , вводится вспомогательный *оператор переименования атрибутов*. В результате применения операции переименования атрибутов получаем новый класс, с измененными именами атрибутов. Оператор переименования атрибутов имеет следующий синтаксис:  $CO.RENAME(R, X, Y, \dots, NX, NY, \dots)$ , где  $R$  - класс,  $X, Y, \dots$  - исходные имена атрибутов,  $NX, NY, \dots$  - новые имена атрибутов.

### Теоретико-множественные операторы

**Определение 2.** *Объединением* двух совместимых по типу коллекций классов  $[A]$  и  $[B]$  называется коллекция классов, содержащая классы того же типа, что и коллек-

ции классов  $[A]$  и  $[B]$ , и элементами-классами, состоящим из классов, принадлежащих или  $[A]$ , или  $[B]$ , или обоим коллекциям. Синтаксис операции объединения:  $CO.UNION([A], [B])$ .

**Определение 3.** *Пересечением* двух совместимых по типу коллекций  $[A]$  и  $[B]$  называется коллекция с тем же набором атрибутов у своих классов, что и у классов, входящих в  $[A]$  и  $[B]$ , и состоящая из классов, принадлежащих одновременно обоим коллекциям  $[A]$  и  $[B]$ . Синтаксис операции пересечения:  $CO.INTERSECT([A], [B])$

**Определение 4.** *Вычитанием* двух совместимых по типу коллекций  $[A]$  и  $[B]$  называется коллекция с тем же набором атрибутов, что и у коллекций  $[A]$  и  $[B]$ , и состоящая из классов, принадлежащих коллекции  $[A]$  и не принадлежащих коллекции  $[B]$ . Синтаксис операции вычитания:  $CO.MINUS([A], [B])$ .

**Определение 5.** Декартовым произведением двух коллекций классов  $[A](A1, A2, \dots, An)$  и  $[B](B1, B2, \dots, Bm)$  называется коллекция, состоящая из классов, являющихся сцеплением классов коллекций  $[A]$  и  $[B]$ :  $(A1, A2, \dots, An, B1, B2, \dots, Bm)$ , таких, что  $(A1, A2, \dots, An) \in [A]$ ,  $(B1, B2, \dots, Bm) \in [B]$ , а атрибуты классов результирующей коллекции являются *сцеплением атрибутов классов*, содержащихся в  $[A]$  и  $[B]$   $(a1, a2, \dots, an, b1, b2, \dots, bm)$ . Синтаксис операции декартового произведения:  $CO.TIMES([A], [B])$ . Практически класс результирующей коллекции появляется в результате множественного наследования от соответствующих классов исходных коллекций.

### Специальные операции

**Определение 6.** *Выборкой (ограничением, селекцией)* на коллекции  $[A]$  с условием  $\lambda$  называется коллекция того же типа классов, что и у коллекции  $[A]$ , состоящая из классов, значения атрибутов которых при подстановке в условие  $\lambda$  дают значение ИСТИНА.  $\lambda$  представляет собой логическое выражение, в которое могут входить атрибуты классов коллекции  $[A]$  и (или) скалярные выражения.

В простейшем случае условие  $\lambda$  имеет вид  $X\Theta Y$ , где  $\Theta$  - один из операторов сравнения ( $=, \neq, <, \leq, >, \geq$ ), а  $X$  и  $Y$  - атрибуты класса коллекции  $[A]$  или скалярные значения. Такие выборки называются  $\Theta$ -*выборки (эта-выборки)* или  $\Theta$ -*ограничения,  $\Theta$ -селекции*. Синтаксис операции выборки:  $CO.WHERE([A], \lambda)$ , или  $CO.WHERE([A], X\Theta Y)$ .

Смысл операции выборки очевиден - выбрать классы коллекции, удовлетворяющие некоторому условию. Таким образом, операция выборки дает «горизонтальный срез» коллекции по некоторому условию.

**Определение 7.** *Проекцией* коллекции  $[A]$  по атрибутам  $X, Y, \dots, Z$ , где каждый из атрибутов принадлежит классам коллекции  $[A]$ , называется коллекция с атрибутами классов  $(X, Y, \dots, Z)$ , содержащая множество классов с атрибутами вида  $(x, y, \dots, z)$ , таких, для которых в коллекции  $[A]$  найдутся классы со значением атрибута  $X$  равным  $x$ , значением атрибута  $Y$  равным  $y$ , ..., значением атрибута  $Z$  равным  $z$ . Синтаксис операции проекции:  $CO.PROJECTION([A], X, Y, \dots, Z)$ .

Операция проекции дает «вертикальный срез» коллекции, в котором удалены все возникшие при таком срезе дубликаты классов.

Обычно рассматривается несколько разновидностей операции соединения. Все разновидности соединения являются частными случаями общей операции.

**Определение 8.** Соединением коллекций  $[A]$  и  $[B]$  по словию  $\lambda$  называется коллекция  $CO.TIMESWHERE([A], [B], \lambda)$ ,  $\lambda$  представляет собой логическое выражение, в которое могут входить атрибуты коллекций  $[A]$  и  $[B]$  и (или) скалярные выражения.

**Определение 9.** Пусть коллекция  $[A]$  содержит атрибут  $X$ , коллекция  $[B]$  содержит атрибут  $Y$ , а  $\Theta$  - один из операторов сравнения ( $=, \neq, <, \leq, >, \geq$  и т.д.). Тогда  $\Theta$ -соединением коллекции  $[A]$  по атрибуту  $X$  с коллекцией  $[B]$  по атрибуту  $Y$  называют коллекцию  $CO.TIMESWHERE([A], [B], X\Theta Y)$ .

Наиболее важным частным случаем  $\Theta$ -соединения является случай, когда  $\Theta$  есть равенство. Синтаксис экви-соединения:  $CO.TIMESWHERE([A], [B], X=Y)$ . Недостатком экви-соединения является то, что если соединение происходит по атрибутам с одинаковыми наименованиями, то в классах результирующей коллекции появляется два атрибута с одинаковыми значениями. Избавиться от этого недостатка позволяет естественное соединение.

**Определение 10.** Пусть даны коллекции  $[A](A1, A2, \dots, An, X1, X2, \dots, Xp)$  и  $[B](X1, X2, \dots, Xp, B1, B2, \dots, Bm)$ , имеющие у своих классов одинаковые атрибуты  $X1, X2, \dots, Xp$  (т.е. атрибуты с одинаковыми именами и одинаковыми типами). Запись типа  $[A](X1, X2, \dots, Xn)$  означает, что коллекция  $[A]$  содержит классы с атрибутами  $X1, X2, \dots, Xn$ .

Тогда естественным соединением коллекций  $[A]$  и  $[B]$  называется коллекция с атрибутами  $(A1, A2, \dots, An, X1, X2, \dots, Xp, B1, B2, \dots, Bm)$  и содержащая множество классов с соответствующими атрибутами  $(a1, a2, \dots, an, x1, x2, \dots, xp, b1, b2, \dots, bm)$ , таких, что  $(a1, a2, \dots, an, x1, x2, \dots, xp) \in [A]$  и  $(x1, x2, \dots, xp, b1, b2, \dots, bm) \in [B]$ . Естественное соединение настолько важно, что для него используют специальную операцию:  $CO.JOIN([A], [B])$ .

Можно выполнять последовательное естественное соединение нескольких коллекций. Нетрудно проверить, что соединение обладает свойством ассоциативности, т.е.  $CO.JOIN(CO.JOIN([A], [B]), [C]) = CO.JOIN([A], CO.JOIN([B], [C]))$ . Поэтому такие соединения можно записывать, опуская скобки:  $CO.JOIN([A], [B], [C])$ .

**Определение 11.** Пусть даны коллекции  $[A](X1, X2, \dots, Xn, Y1, Y2, \dots, Ym)$  и  $[B](Y1, Y2, \dots, Ym)$ , причем атрибуты  $Y1, Y2, \dots, Ym$  - общие для двух коллекций. Делением коллекций  $[A]$  на  $[B]$  называется коллекция с набором атрибутом у классов  $(X1, X2, \dots, Xn)$  и содержащая множество атрибутов классов  $(x1, x2, \dots, xn)$ , таких, что для всех атрибутов классов  $(y1, y2, \dots, ym) \in [B]$  в коллекции  $[A]$  найдется атрибут класса  $(x1, x2, \dots, xn, y1, y2, \dots, ym)$ .

Коллекция  $[A]$  выступает в роли делимого, коллекция  $[B]$  выступает в роли делителя. Синтаксис операции деления:  $CO.DEVIDBY([A], [B])$ .

### Соответствие операторов реляционной и объектно-реляционной алгебр

Проведем соответствие теоретико-множественных операторов реляционной и ОРА алгебр. Введем  $f_{ORA}(R)$  оператор преобразования реляционного оператора в объектную форму, где  $R$  - реляционное отношение.  $f_{REL}(C)$  - оператор преобразования объектной формы оператора в реляционный вид, где  $C$  - класс. С помощью этих функ-

ций мы можем провести параллель между объектной и реляционной моделью. Например, для операции Объединение.  $f_{ORA}(A \text{ UNION } B) = CO.UNION([A], [B])$ .  $f_{REL}(CO.UNION([A], [B])) = A \text{ UNION } B$ .

### Особые географические операторы

В дополнение к стандартным операторам, набор пространственных связей может быть проверен между двумя классами, имеющими географическое отображение, основным и дополнительным. Все операции происходят с помощью специального класса COG.

**Равенство.** Эта функция класса возвращает истину, если вторая фигура идентична основному объекту. Две фигуры равны, если они имеют ту же координатную систему, тот же тип, одинаковое число точек и одинаковые точки в одинаковом порядке ( $x, y, z$  и мера).  $COG.EQUALS(A, B)$ .

**Касание.** Эта функция возвращает истину, если две фигуры делят общую границу.  $COG.TOUCHES(A, B)$ .

**Внутри.** Эта функция возвращает истину, если основная фигура полностью содержится во второй фигуре.  $COG.WITHIN(A, B)$ .

**Содержит.** Эта функция возвращает истину, если вторая фигура полностью входит в основную. Это тоже самое, что и поменять местами основную и дополнительную фигуру с оператором Внутри.  $COG.CONTAINS(A, B)$ .

**Скрещивание.** Эта функция возвращает истину, если пересечение внутренних областей двух фигур не пусто.  $COG.CROSSES(A, B)$ .

**Частичное совпадение.** Эта функция возвращает истину, если результатом пересечения двух фигур есть объект той же размерности, но отличный от обоих фигур. Подходит для вариантов типов область/область, линия/линия и множество точек/множество точек.  $COG.OVERLAPS(A, B)$ .

**Нет пересечения.** Эта функция возвращает истину, если две фигуры не пересекаются.  $COG.DISJOINT(A, B)$ .

**Наложение.** Эта функция соединяет две фигуры. Возвращает остаток основной фигуры и остаток дополнительной, с содержанием любого куска, который входит только в основную или дополнительную фигуру.  $COG.OVERLAY(A, B)$ .

**Пересечение.** Эта функция поиска возвращает набор пересечений основной и дополнительной фигуры (пространственное логическое «И»), которые общие для обеих фигур.  $COG.INTERSECT(A, B)$ .

**Разность.** Эта функция поиска возвращает разность между основной и дополнительной фигурой (пространственное логическое отрицание И).  $COG.DIFFERENCE(A, B)$ .

**Симметрическая разница.** Эта функция поиска возвращает симметрическую разницу основной и дополнительной фигуры (логическое пространственное исключающее «ИЛИ»). Возвращает разницу обеих, основной и дополнительной фигуры.  $COG.SYMMETRICALDIFFERENCE(A, B)$ .

**Объединение.** Эта функция поиска возвращает объединение основной и дополнительной фигуры (логическое пространственное «ИЛИ»).  $COG.UNION(A, B)$ .

**Функция отсечения.** Эта операция возвращает порцию основной фигуры, которая находится внутри заданной границы. Действует для всех типов фигур.  $COG.CLIPFUNCTION(A, B)$ .

**Зависимые и примитивные операторы**

Не все операторы ОРА являются независимыми - некоторые из них выражаются через другие ОРА операторы. Оператор соединения определяется через операторы декартового произведения и выборки. Для оператора естественного соединения добавляется оператор проекции. Оператор пересечения выражается через вычитание. Оператор деления выражается через операторы вычитания, декартового произведения и проекции. Таким образом, показано, что операторы *соединения, пересечения и деления* можно выразить через другие ОРА операторы, т.е. эти операторы не являются примитивными. Оставшиеся ОРА операторы (*объединение, вычитание, декартово произведение, выборка, проекция*) являются *примитивными операторами* - их нельзя выразить друг через друга. Оператор декартового произведения - это единственный оператор, *увеличивающий количество атрибутов*, поэтому его нельзя выразить через объединение, вычитание, выборку, проекцию. Оператор проекции - единственный оператор, *уменьшающий количество атрибутов*, поэтому его нельзя выразить через объединение, вычитание, декартово произведение, выборку. Оператор выборки - единственный оператор, позволяющий проводить сравнения по атрибутам коллекции, поэтому его нельзя выразить через объединение, вычитание, декартово произведение, проекцию.

**Метод преобразования**

Запишем связь в виде формулы, которая будет выглядеть так:

$$A(< A_1 : D_1 >, < A_2 : D_2 >, \dots < A_n : D_n >) \Leftrightarrow \begin{cases} A'.A_1 : D'_1 \\ A'.A_2 : D'_2 \\ \dots \\ A'.A_n : D'_n \end{cases},$$

где

$A$  - реляционное отношение,  
 $A_1, A_2, \dots, A_n$  - атрибуты отношения,  
 $D_1, D_2, \dots, D_n$  - домены соответствующих атрибутов отношения,

$A'$  - класс, представляющий собой отображение отношения  $A$  в объектную модель,

$A'.A_1, A'.A_2, \dots, A'.A_n$  - свойства класса  $A'$ , которые являются преобразованными атрибутами реляционного отношения  $A$ ,

$D'_1, D'_2, \dots, D'_n$  - типы данных, соответствующие доменам атрибутов отношения  $A$ . При преобразовании необходимо преобразовывать типы данных. Введем функцию преобразования типов:

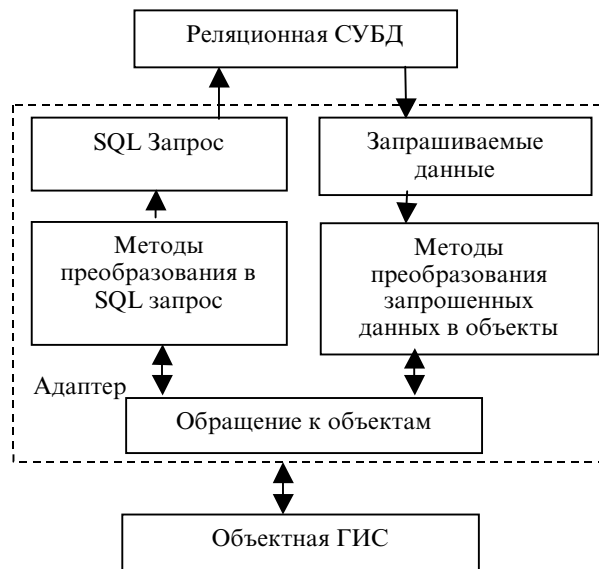
$$\begin{aligned} f_{rto}(D_1, D_2, \dots, D_n) &\Rightarrow D'_1, D'_2, \dots, D'_n \\ f_{otr}(D'_1, D'_2, \dots, D'_n) &\Rightarrow D_1, D_2, \dots, D_n \end{aligned},$$

где

$f_{rto}$  - функция преобразования доменов реляционного отношения в типы используемые объектом,

$f_{otr}$  - функция преобразования типов, которые используются объектом в домены реляционного отношения.

Функциональная схема сопряжения изображена на рисунке ниже.



**Рисунок 2.** Метод функционирования объектно-реляционного сопряжения

Новизна метода состоит в изменении принципа сопряжения. Это не обработка запросов от ГИС, а преобразование реляционной информации в объектную, чтобы ГИС обращалась к любым элементам пространственного комплекса как к объекту с определенным набором свойств. Генерация запросов при обращении ГИС к объекту лежит на адаптере. Метод позволяет при построении ГИС полностью абстрагироваться от реляционной модели данных и работать только с объектами.

**Выводы**

В статье описана разработанная объектно-реляционная алгебра, предназначенная для сопряжения объектно-ориентированной ГИС и РБД. ОРА алгебра представляет собой набор операторов, использующих коллекции классов в качестве аргументов, и возвращающие коллекции классов в качестве результата. ОРА алгебра замкнута таким образом, что результаты одних ОРА выражений можно использовать в других выражениях. Определены двадцать один оператор, которые объединены в три группы: теоретико-множественные операторы, специальные ОРА операторы и особые географические операторы. Использование возможностей, предоставляемой ОРА, позволит сопрягать ГИС и РБД более эффективно, чем в технологии ArcSDE. Программный адаптер сопряжения, реализующий сопряжение на основе ОРА, позволяет повысить скорость разработки ГИС. Адаптер внедрен в ряде организаций.

**Литература**

- 1) Date. С. J. The relational model will stand the test of time // Intelligent Enterprise, 1999. Vol. 2, No 8.
- 2) Андреев А., Березкин Д., Самарев Р. Внутренний мир объектно-ориентированных СУБД // Открытые системы. - 2001. - №3. С. 34 - 37.
- 3) Буч Г. Объектно-ориентированное проектирование с примерами применения М., «Конкорд», 1992. - 345 с.
- 4) Кирстен В., Ирингер М. и др. СУБД Cache. Объектно-ориентированная разработка приложений. СПб.: Питер, 2001. - 204с.

5) Конноли Т., Берг К. Базы Данных: проектирование, реализация и сопровождение. Теория и практика. - М.: Издательский дом «Вильямс», 2000. - 1120 с.

6) Ляховец С.В. Сопряжение объектно-ориентированной ГИС с реляционной базой данных // Международная научная конференция «Теория и техника передачи, приема и обработки информации»: Сб. научных трудов. - Харьков: ХНУРЭ, 2003г., С.268-269.

7) Ляховец С.В., Четвериков Г.Г. Геоинформационные системы для задачи управления проектами сложных природно-технических комплексов // Радиоэлектроника и информатика. - Харьков: ХТУРЭ. - 2001. - № 3. - С. 114 - 121.

8) Ляховец С.В., Четвериков Г.Г. Концепции модели данных для трубопроводов // 7-я Международная конференция «Теория и техника передачи, приема и обработки информации»: Сб. научных трудов. - Харьков: ХТУРЭ, 2001. С.267-268.

9) Мейер М. Теория реляционных баз данных. - М.: Мир, 1987. - 608 с.

10) Стрельцов И. Технология ArcSDE - что это // ARCREVIEW. - 2001. - №4. - С. 11-12.

УДК 574:504.064

## КОМПЛЕКСНЫЕ РЕШЕНИЯ ОРГАНИЗАЦИИ ИНТРАСЕТЕВЫХ ТЕХНОЛОГИЙ ЭКОЛОГИЧЕСКОГО МОНИТОРИНГА (ЧАСТЬ 1)

*Н.А. Любимова*

*Кандидат технических наук. Преподаватель НВК №55.*

*Л.В. Шандыба*

*Аспирант.*

*Кафедра автоматики и радиоэлектроники Украинской инженерно-педагогической академии (УИПА), ул. Университетская, 16, г. Харьков, Украина, 61003  
Контактный телефон: +38(0572) 26-42-66  
e-mail: shandiba@rambler.ru*

*В данной статье предлагаются этапы внедрения комплексных решений организации интрасетевых технологий экологического мониторинга. Рассматриваются задачи и концепции разработки проекта с использованием спиральной модели SSADM. Приводится общая функциональная карта использования и внедрения предлагаемых решений.*

### 1. Введение

Экологический мониторинг природной среды предполагает обычно пассивное наблюдение за объектами, например, гидроэкосистемы. Однако, в настоящее время практически во всех ведущих странах мира интенсивно разрабатываются *методы* так называемого *активного контроля* за качеством водной среды с использованием, например, методов биотестирования [1,2]. Применение биологических тест-объектов в качестве первичных преобразователей измерительных систем и высокочувствительных методов исследования реакции последних на внешние воздействия позволяют создать специализированные интрасетевые информационно-аналитические системы по оценке степени антропогенного воздействия на модельные и естественные биологические системы. Целью данной работы является создание организационной модели интрасетевой технологии экологического мониторинга.

### 2. Основное содержание работы, используемые методы

Программное обеспечение комплекса строится с учетом поддержки различных платформ, *объектных технологий OLE, ActiveX, CORBA, технологий "клиент-сервер"* и *распределенных сетевых вычислений*. Пред-

лагаемые комплексные решения по внедрению интрасетевых технологий такого назначения базируются на следующих этапах:

- Создание локальных, региональных и глобальных систем агроэкологического мониторинга, включающих сеть опорных пунктов с унифицированными методами диагностики, набором нормативных и правовых документов, разветвленной структурой, единой базой обработки результатов и стандартизированным набором биологических тест-объектов.

- Объединение опорных пунктов в единую сеть для организации распределенного доступа к единым базам и хранилищам данных, проведение аналитических, прогнозических исследований, моделирование и соответствующая корректировка параметров технологических процессов вредного производства.

- Гибкое внедрение, полная адаптация и объединение с существующими на предприятии информационными системами. Включение в логические модели данных и структуры запросов дополнительной информации, участвующей в управлении технологическим процессом.

- Использование современных сетевых, клиент-серверных, объектных технологий доступа и организации хранилищ данных, обеспечение гибкой системы отчетности и обработки многомерных запросов к хранилищам данных.

- Полное сопровождение и обслуживание внедренных систем, обучение персонала. Проведение предпроектных исследований и дополнение внедряемых техноло-