

А. Н. Родионов, д-р техн. наук, вед. науч. сотр. e-mail: ran@newmail.ru,  
Вычислительный центр ДВО РАН, г. Хабаровск

## Периодические данные: идентификация, моделирование и представление в базах данных

*Рассматривается полный спектр вопросов моделирования данных, повторяющихся с определенной периодичностью. В первую очередь решается задача по их идентификации. Строится модель, на основании которой могут быть определены параметры циклических процессов, являющихся источником таких данных. Вводятся и исследуются на предмет сопоставимого качества моделирующие структуры и соответствующие им реляционные отношения. Завершает работу рассмотрение вопросов привязки моделируемых структур к единому модельному времени баз данных и перечисление направлений перспективных исследований в этой области.*

**Ключевые слова:** периодические и непериодические циклические данные, тип, экземпляр и клон сущности, многофункциональные циклические структуры данных, темпоральные базы данных

### Введение

Конструирование структур, в которых будут храниться данные о всевозможных "расписаниях" (графиках, планах и подобных им по своему назначению объектах), — одна из типичных задач даталогического моделирования. По сути, в подобных объектах сосредоточена информация о прогнозируемом поведении сущностей. В этих же объектах достаточно часто встречаются как данные, повторяющиеся с некоторой заданной периодичностью (далее "периодические циклические" данные — *pd*), так и данные с произвольным периодом повторения ("непериодические циклические" данные — *nd*).

О наличии или отсутствии *pd* могут косвенно свидетельствовать и ограничения, действующие в тех или иных предметных областях. Вот только два примера подобных ограничений.

*Ограничение 1.* "Лекции по определенной дисциплине читаются один раз в неделю с продолжительностью 2 академических часа".

*Ограничение 2.* "Рейс номер такой-то выполняется 2 раза в неделю по понедельникам и пятницам".

*Лекции и рейсы* здесь претенденты на *pd*, поскольку имеет место повторение и первых, и вторых через равные промежутки времени.

Цикличность данных (если она установлена), период повторения, а также факт идентификации повторяющихся данных дают возможность компактно (используя минимальное число отношений и короткей) представить "расписания" в базах данных.

Разработка схем отношений для простейших "расписаний", когда известны все перечисленные вы-

ше характеристики, не составляет большого труда. В действительности же зачастую только выделить "циклические данные" из массивов документов (которые являются исходным материалом для проектирования) не всегда удается "с первого раза". Например, *лекции* и *рейсы* были отнесены к разряду *pd* скорее интуитивно, чем формально, потому что в обоих случаях негласно присутствуют связанные с ними агенты — дисциплины и воздушные суда соответственно.

Не все очевидно и в отношении периода повторения, например *рейсов*, так как последние выполняются (если судить по ограничению) не только 2 раза в неделю, но и в определенные дни недели. Очевидно, что в последнем случае фиксация в базе данных подобного рода сведений обязательна и "недельный" период требует отдельной, как одно из решений, двукратной привязки к порядковому номеру дня.

Трудности моделирования "циклических" данных обусловлены не только сложностями их обнаружения, формализации и идентификации, но и необходимостью учета времени как фактора их существования. Научное сообщество последнее десятилетие активно развивает темпоральные аспекты концептуального моделирования данных. Они охватывают широкий спектр направлений, включая вопросы отражения траекторий и семантики "движущихся" объектов [1], разработки специфических приемов представления периодичности в виде повторяющихся квантов времени [2], построения и определения областей применения специальных моделей

времени [3, 4], к которым "привязываются" объекты моделей данных. Просматривается тенденция на стандартизацию и ограничение числа базовых временных категорий с унификацией алгоритмов их сопряжения с данными [5]. В то же время аспекты моделирования повторяющихся (циклических) данных, отражающих прогнозные траектории, практически выпадают из поля зрения исследователей.

В работе рассматривается весь комплекс вопросов моделирования, начиная с идентификации собственно циклических данных и заканчивая конструированием и использованием структур, предназначенных для размещения *pd* в базах данных реальных систем.

### "Циклические" данные.

#### Определение и вопросы моделирования

Конечная цель любого моделирования данных — получение качественных концептуальных схем. Качество — комплексная категория и применительно к данным включает множество составляющих, среди которых функциональность (полнота), избыточность и отсутствие аномалий [6, 7], равно как и гибкость, простота в использовании, эффективность [8], а также компактность [9], считаются наиболее важными.

При рассмотрении и анализе "циклических" структур ограничимся единственной качественной характеристикой — компактностью, полагая, что соответствие первым трем по умолчанию всегда должно быть достигнуто, а гибкость, простота в использовании и эффективность требуют расширения рамок моделирования, поскольку на их значении оказывают влияние и другие части информационной системы.

Компактными будем полагать те структуры (если более точно, то соответствующие им отношения), которые, если сравнивать их с альтернативными конструкциями, характеризуются меньшей мощностью и разреженностью. (В дальнейшем, при переходах с концептуального уровня моделирования на логический будем ориентироваться на реляционные модели данных, используя соответствующие терминологию и структурную организацию последних.)

Определившись с основным "качественным" критерием, исследуем природу и источники периодических данных в моделируемых предметных областях (универсумах).

На интуитивном уровне "циклические" данные всегда ассоциируются с данными, повторяющимися с некоторой регулярностью. Для их обнаружения в конкретных универсумах сказанного недостаточно. Нужно иметь более точное, формальное определение. Для его получения будем придерживаться основополагающих моментов, касающихся как наиболее общего взгляда на базу данных и ее организацию, так и на устройство организационных систем,

в которых *pd* можно обнаружить наиболее часто. (Ссылка на организационные системы сделана из тех соображений, что при планировании какой-либо деятельности специалисты стремятся к тому, чтобы она протекала по возможности ритмично.)

Поскольку любая предметная область с точки зрения моделирования данных есть не что иное, как набор взаимодействующих между собой сущностей, включая сущности, поступающие извне, сущности, потребляемые и производимые универсумом, а также сущности, покидающие его границы, то база данных, будучи нацеленной на отражение перечисленных аспектов, должна содержать статические и динамические данные, ассоциированные исключительно с сущностями. Фиксируемые в базе данных сведения (факты) могут касаться как текущих моментов времени  $t_c$ , так и будущих периодов. Ретроспективные данные вносятся в базу данных, как правило, при ее создании и играют роль своеобразных начальных условий, отражающих состояние универсума в определенный момент времени  $t_s \leq t_c$ . Текущие данные становятся ретроспективными при переходе к новому транзакционному периоду. Статические базы данных именуются традиционными, а динамические, включающие в том числе обязательные статические аспекты, — темпоральными. Для отражения динамики сущностей ("изменений" сущностей) в базе данных необходимо сосредоточить все факты, касающиеся сведений о местоположениях, состояниях и ролях сущностей.

Из вышеизложенного следует, во-первых, что любое "циклическое данное" должно всегда ассоциироваться с сущностью (принадлежать ей), поскольку ничего, кроме сущностей и характеристик последних, база данных по определению содержать не может. Во-вторых, сама ссылка на "циклическость" указывает на то, что такие данные могут появляться только в темпоральных базах данных, и, следовательно, изменения следует рассматривать в контексте "процессов", что в реальности и происходит. Будем поэтому далее увязывать "циклические" данные с парой <тип сущности, тип процесса> ( $\langle e, p \rangle$ ). Выбор подобного составного объекта для идентификации *pd* предопределил все последующие теоретические и практические построения.

Кроме традиционных понятий, используемых в даталогическом моделировании, таких как тип и экземпляр [10], позволяющих различать именованные множества и их отдельные экземпляры, для целей настоящего исследования несколько расширим терминологический базис, добавив в него новую категорию "клон" как совокупность идентифицируемых идентичных экземпляров сущностей. С помощью "клонов" будем отличать отдельные, одинаковые экземпляры в последовательностях, которые эти экземпляры будут образовывать.

Если, например, в предметной области "Авиационное сообщение" "рейсы" — это тип, "рейс

№ 8457" — экземпляр, принадлежащий типу "рейс", то "рейс № 8457", выполняемый по вторникам и четвергам", — два клона "рейса № 8457".

Сказанное позволяет в первом приближении определиться со схемой (заголовком) отношения, в котором будут размещаться  $pd$ :

$$R(\text{Цикле}_{ij}) = \{\bar{E}_i^*, \bar{P}_j^*, \bar{T}^*, \bar{r}\}, \quad (1)$$

где  $i$  — индекс типа сущности;  $j$  — индекс типа процесса;  $\bar{E}_i^*$  — атрибут, идентифицирующий экземпляр сущности  $i$ -го типа;  $\bar{P}_j^*$  — атрибут, идентифицирующий процесс  $j$ -го типа;  $\bar{T}^*$  — ссылка на временной период;  $\bar{r}$  — атрибут, "указывающий" на число повторений; надстрочный индекс \* — признак ключевого атрибута. Элементы последнего столбца в отношении, которое соответствует схеме (1), будут содержать непосредственно  $pd$ .

Обозначим круг задач, возникающих при конкретизации и расширении полученного отношения, пригодного для моделирования  $pd$  уже реальных систем.

Обратим внимание на то, что конечной целью является получение моделирующих структур, наилучшим образом, с позиции качества, представляющих  $pd$  в базах данных. В связи с этим первоочередная задача — поиск, идентификация и нахождение параметров  $pd$ , актуальных для моделируемого универсума. Для ее решения потребуется построить формальную модель циклических процессов и разработать на ее основе соответствующий алгоритм.

Следующий шаг — подбор эффективных моделирующих структур, позволяющих фиксировать всевозможные параметры периодичности, и оценка качества таких структур.

Решение проблемы сочетания повторяющихся периодических и непериодических данных, что часто встречается на практике, составляет содержание еще одной задачи моделирования  $pd$ .

Общесистемные аспекты охватывают обязательные вопросы привязки циклических траекторий к осям реального и модельного времени, а также решения по организации хранилищ ретроспективных данных.

В представленной последовательности будет строиться все дальнейшее изложение.

### Модели циклических процессов

Детализации схемы отношения (1) должны предшествовать рассмотрение и формализация по возможности всех наблюдаемых вариантов периодичности, которые встречаются на практике.

Периодичность в информатике, если подходить к этому вопросу формально, будет означать то же самое, что и в математике. Должно существовать число  $m \neq 0$  такое, чтобы выполнялось условие  $f(x + m) = f(x) = y$ . Поскольку  $pd$  ассоциируются с парой  $\langle e, p \rangle$ , то повторяться с известной регулярностью при фиксированном  $e$  должны про-

цессы. Следовательно, циклическую функцию применительно к рассматриваемой задаче можно записать в виде

$$f(t_{pj} + T_{pj}) = f(t) = p_{pj}$$

где  $t_{pj}$  — текущий момент времени начала экземпляра процесса  $p_{pj}$  (подстрочный индекс  $p$  идентифицирует экземпляр процесса  $j$ -го типа);  $T_{pj}$  — продолжительность периода повторения  $p_{pj}$ .

Таким образом, моделирование, по сути, сводится к нахождению  $t_{pj}$  и  $T_{pj}$  для всех экземпляров процессов  $p_{pj}$ , принадлежащих некоторому типу  $P_j$ . (Обозначение  $P_j$  введено, чтобы различать атрибут  $\bar{P}_j$  в отношении (1) и множество процессов  $P_j$ . Далее, моделируя циклические экземпляры  $p_{pj}$ , индекс  $j$  будем опускать, полагая, что все  $p_p$  принадлежат конкретному  $P_j$ .)

Отметим, что каждое множество  $P_j$  характеризуется набором уникальных параметров, часть из которых идентифицирует экземпляры  $p$ , а часть — свойства процесса. Отсюда, отдельные  $p_p$  различаются значениями, принимаемыми параметрами, образующими вышеупомянутые группы.

Для определения искомого  $T_p$  воспользуемся графическими моделями, отражающими существенные аспекты периодичности. Простейшая из них показана на рис. 1.

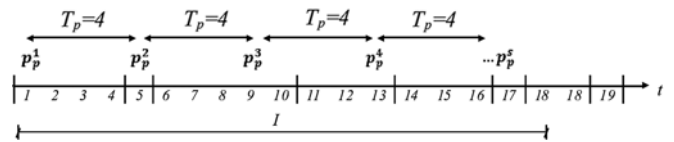


Рис. 1. Сеть единственного циклического процесса

Клоны  $p_p$  образуют конечную последовательность  $\{p_p^s\}$ , где  $s$  — порядковый номер элемента последовательности. Некоторый обозримый период времени  $I$ , в котором ведется поиск повторяющихся процессов, разбит на упорядоченные интервалы  $c$  равной длины, именуемые квантами. Каждому кванту поставлен в соответствие порядковый номер  $m$ , позволяющий как идентифицировать кванты, так и различать предшествующий и последующий кванты при их сравнении. Другими словами, имеет место последовательность  $\{c_m\}$ . Начало любого клона  $p_p$  всегда ассоциируется с конкретным  $c_m$ . Таким образом, можно говорить об отображении  $p_p \rightarrow c$ . Периодическая циклическость (далее просто циклическость) в интервале  $I$  для  $p_p$  будет иметь место в том случае, если для  $\forall \text{пар} \langle p_p^{s+1}, p_p^s \rangle$ , где  $s = \overline{1, S-1}$ , выполняется условие:  $(c_m^{s+1} - c_m^s) = T_p$ . (Здесь в соответствии с ранее введенным отображением  $c_m^{s+1} = f(p_p^{s+1})$ , а  $c_m^s = f(p_p^s)$ .)

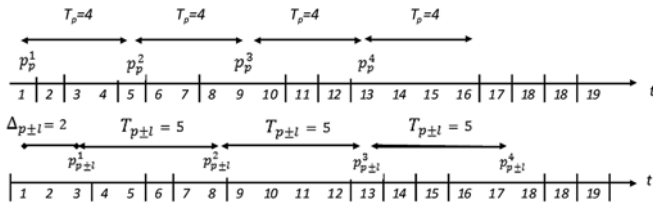


Рис. 2. Сети циклических процессов со смещением и без смещения

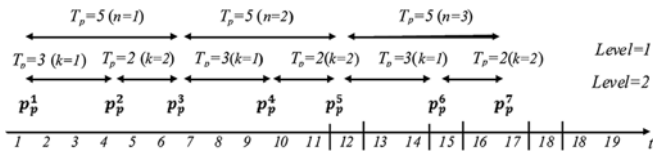


Рис. 3. Множественная (групповая) цикличность для группы из двух клонов

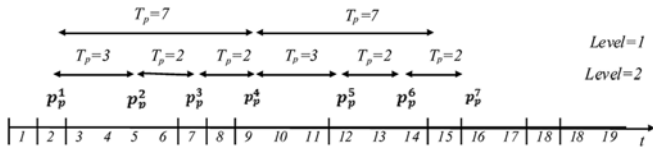


Рис. 4. Множественная цикличность для группы из трех клонов

Понятно, что начало  $p_p^1$  не всегда совпадает с  $c_1$ .

Более того, и другие  $p_p^l$ , где  $l$  — натуральное число, могут стартовать в собственные  $c_m^l$  (рис. 2). Поэтому имеет смысл для каждого  $p_p$  фиксировать смещение начала  $\Delta_p$  первого клона  $p_p^1$  от начала координат.

Рассмотренная цикличность не является единственной. Общий случай, когда повторяются группы процессов, показан на рис. 3 и 4. Период  $T_p$  для

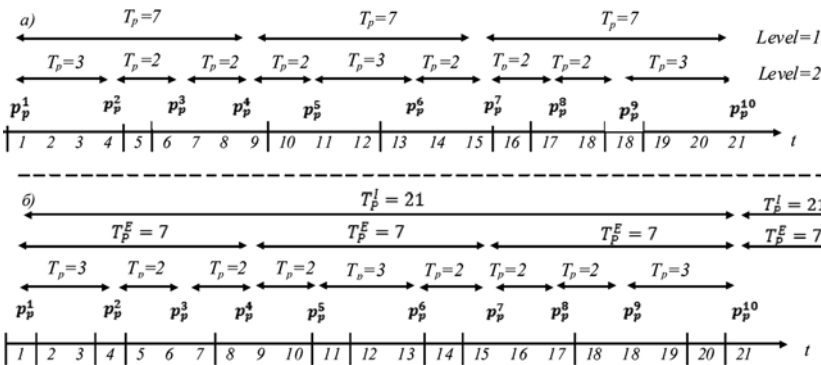


Рис. 5. Групповая цикличность

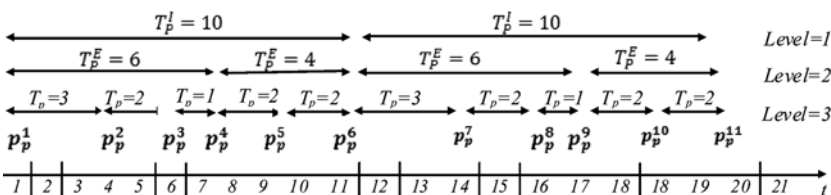


Рис. 6. "Шаговая" периодическая цикличность

группы из двух процессов (рис. 3) составляет 5 квантов, трех процессов (рис. 4) — 7 квантов.

Заметим, что для существования периодичности должно выполняться условие, несколько отличное от сформулированного ранее. Если более точно, то речь идет о комплексе условий. Применительно к ситуации, представленной на рис. 3, эти условия следующие:

$$\begin{cases} (c_5 - c_2) = (c_{10} - c_7) = (c_{15} - c_{12}); \\ (c_7 - c_5) = (c_{12} - c_{10}) = (c_{17} - c_{15}). \end{cases}$$

Для случая, показанного на рис. 4, должны выполняться условия

$$\begin{cases} (c_5 - c_2) = (c_{12} - c_9); \\ (c_7 - c_5) = (c_{14} - c_{12}); \\ (c_9 - c_7) = (c_{16} - c_{14}). \end{cases}$$

Опираясь на приведенные примеры, запишем в общем виде условие групповой цикличности, которое для  $\forall$  периодов  $n$ , содержащих одинаковое  $k$ -е число процессов, и фиксированных  $k$  имеет вид

$$f(p_p^{n(k+1)}) - f(p_p^{nk}) = \text{const}. \quad (2)$$

Нельзя оставить без внимания ситуацию, когда условие (2) не соблюдается, но группы процессов, тем не менее, повторяются с известной регулярностью (рис. 5). С периодичностью  $T_p = 7$  повторяются группы, состоящие из трех клонов.

Тем не менее, можно предположить, что при значительном увеличении продолжительности анализируемого периода  $I$  (как минимум в два раза) искомая периодичность будет обнаружена. Пример подобного рода приводится на рис. 5, б. Пусть повторяются группы из девяти клонов с интегральным (действительным) периодом  $T_p^I$ , равным 21-му кванту. При этом условие (2) выполняется и, следовательно, присутствует периодическая цикличность.

Чтобы различать периоды  $T_p^I$  и  $T_p^E$ , последний будем именовать локальным периодом повторения, а соответствующую последовательность принадлежащих ему клонов — вложенной последовательностью (по аналогии с вложенными циклами, используемыми в программировании).

В завершение рассмотрения вариантов групповой цикличности приведем пример вложенной последовательности, элементы которой повторяются с некоторым шагом (смещением). Величина последнего в общем случае определяется как  $\sum_{j=1 \div N} T_p^j$ , где  $j$  — номер последовательности, следующей за искомой (рис. 6). Здесь последовательность из трех клонов повторяется с шагом  $\sum_{j=1 \div 1} T_p^j = 4$ .

Можно предположить, что существует иерархия групп со сколь угодно большим числом уровней иерархий *Level*, каждый из которых может включать вложенные последовательности как с нулевым шагом, так и с шагом, отличным от нуля. Но это предположение легко опровергается, поскольку все промежуточные уровни — это не более чем манипуляции с разбиением одного единственного базового интегрального уровня, содержащего периоды  $T_p^I$ , на подуровни.

Кроме того, подобная декомпозиция не оказывает никакого влияния на решение поставленной ранее задачи по полному и компактному представлению периодических данных в базах данных, поскольку фиксации подлежат все начальные кванты клонов, образующих интегральную последовательность с периодом  $T_p^I$ .

Рассмотренные комбинации периодической цикличности позволяют сформировать полный набор параметров, которые должны быть представлены в базе данных. Это, кроме очевидных  $T_p$  и  $\Delta_p$ , кванты начала каждого из клонов в процессной группе.

Полученные формальные выражения (1), (2) дают возможность построить однопроходной алгоритм нахождения всех перечисленных параметров для известных последовательностей экземпляров процессов и соответствующих им квантов. (Представленная ниже программа не содержит блока формирования входного массива *Input*, так как это несущественно для рассматриваемого ниже алгоритма.)

Алгоритм последовательно считывает все  $(c_m^{s+1} - c_m^s)$  из массива *Input* и размещает их в двумерном массиве *Output* размерностью  $r \times c$ , где  $r = c = M$ . Каждое очередное значение сначала хранится в переменной *current\_value*, а затем размещается в элементе *Output*, индексируемом текущими значениями  $r$  и  $c$ . При равенстве  $Output[r, c]$  и  $Output[r - 1, c]$  изменяются индексы  $r$  или  $c$ , в противном случае, за счет пересылки на очередные свободные ячейки первой строки данных либо из первого столбца (если  $r > 1$ ) или (при  $r = 1$ ) данных второй строки — формируется последовательность процессов, образующих циклическую группу.

На основании значений, принимаемых переменными  $i$  и  $r$  после завершения работы алгоритма, можно сделать вывод о наличии или отсутствии периодичности, а также получить искомые параметры цикла, если таковой присутствует в анализируемой последовательности, состоящей из  $(c_m^{s+1} - c_m^s)$ . Если  $r > 1$ , то наблюдается цикличность с числом циклов, равным  $r$ . Значение, которое содержится в  $i$ , — не что иное как число элементов (клонов) в циклической группе.

Листинг 1 представляет текст программы на языке C#, реализующий представленный алгоритм.

**Листинг 1.** Расчет параметров цикличности  
 int r = 1, c = 0, i = 1; // r и c — текущие индексы соответственно строки и столбца

```
bool first = true; // массива Output
int[] Input = new int[k]; // i — индекс текущего элемента массива Input
int[,] Output = new int[k, k];
foreach (int current_value in Input)
{ if (first) { Output[0, 0] = Input[0];
  first = false;
} else { Output[r, c] = current_value;
  if (Output[r, c] == Output[r - 1, c])
  { if ((i - c) > 1) c++;
  else { r++; c = 0; } } else { if (r > 1) // Пересылка столбца
  { for (int l = 1; l < r + 1; l++, i++)
  Output[0, l] = Output[l, 0]; r = 1;
  } else // Пересылка строки
  { for (; c >= 0; c--, i++) Output[0, i] =
  Output[l, c];
  c = 0; } } } }
```

При последующем структурном моделировании будем обращать отдельное внимание на случай, когда для какой-то части процессов регулярность (периодичность) отсутствует. Очевидно, что сочетание периодических и непериодических последовательностей процессов также должно быть особым образом отражено на уровне структур данных.

### Моделирующие структуры

В настоящем разделе рассматриваются, анализируются и сравниваются "циклические" и "нециклические" моделирующие структуры, которые могут быть привлечены для размещения в них повторяющихся пар  $\langle e, p \rangle$  соответственно периодических  $pd$  и непериодических  $nd$  данных. В "нециклических" отношениях, в отличие от "циклических", хранятся все клоны экземпляров  $p_p$ . Факт повторяемости клонов в расчет не принимается ввиду отсутствия в их последовательностях постоянного периода повторения.

Типичная "нециклическая" структура в составе кластера, куда вошли также все "родительские" объекты, показана на рис. 7. Непосредственно сами данные о повторяющихся процессах размещаются в *Trivial*-структуре. Порядковые номера квантов, как и, возможно, их продолжительности, если длины всех квантов не равны между собой, хранятся

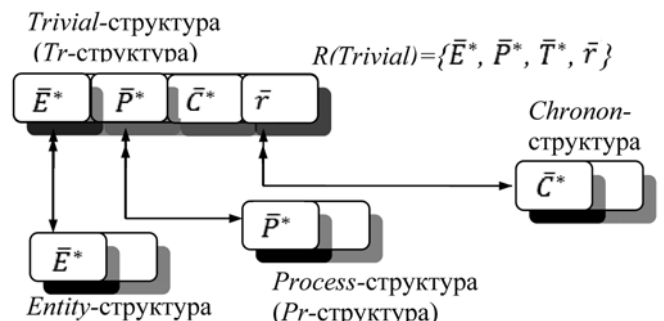


Рис. 7. Организация "нециклического" кластера

в *Chronon*-структуре. Экземпляры сущностей и процессов содержатся соответственно в *Entity*- и *Process*-структурах. (Далее, на части последующих диаграмм, если не требуется раскрывать атрибутивный состав, *Entity*- и *Process*-структуры, включая и связи, ведущие к ним, будут опускаться.)

Универсальная, многоцелевая "циклическая" конструкция представлена на рис. 8. Ее атрибутивный состав позволяет отразить все установленные ранее параметры повторяемости. Обратим особое внимание на атрибут  $\bar{T}_p$  *Entity*-структуры, тем более, что это единственное, что отличает "нециклический" кластер (см. рис. 7) от кластера, в котором учтена повторяемость процессов (см. рис. 8). Соответствующие  $\bar{T}_p$  элементы данных содержат информацию о  $T_p$ . Если более точно — то о длине периода, которая измеряется в квантах, для клонов экземпляра  $p_p$ .

$T_p$  — это расчетная величина и она может быть получена из *Multipurpose*-отношения, если выполняется одно условие. Оно сводится к тому, что в *Multipurpose*-отношении для каждого  $p_p$  должен присутствовать дополнительный кортеж, позволяющий вычислить  $T_p$ . Два примера реляционных отношений, иллюстрирующих сказанное, с использованием атрибута  $T_p$  и без него, для "циклического" кластера продемонстрированы на рис. 9.

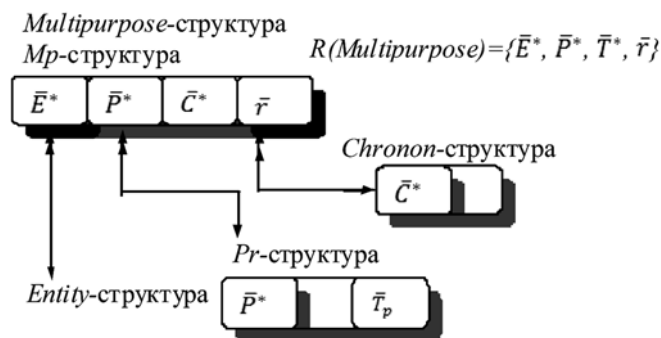


Рис. 8. Структурная организация "циклического" кластера

а) <i>Mp</i> -структура (группа из трех процессов)						<i>Mp</i> -структура (группа из одного процесса)					
$\bar{E}^*$	$\bar{P}^*$	$\bar{C}^*$	$\bar{r}$			$\bar{E}^*$	$\bar{P}^*$	$\bar{C}^*$	$\bar{r}$		
1	1	1	1	$t_1$		1	1	2	1	$t_1$	
1	1	4	1	$t_2$		1	1	4	1	$t_2$	
1	1	6	1	$t_2$							
1	1	8	1	$t_4$							
$T_{p=1} = t_4(\bar{C}) - t_1(\bar{C}) = 7$						$T_{p=1} = t_2(\bar{C}) - t_1(\bar{C}) = 2$					
б) <i>Mp</i> -структура						<i>Mp</i> -структура					
$\bar{E}^*$	$\bar{P}^*$	$\bar{C}^*$	$\bar{r}$			$\bar{E}^*$	$\bar{P}^*$	$\bar{C}^*$	$\bar{r}$		
1	1	1	1	$t_1$		1	1	2	1	$t_1$	
1	1	4	1	$t_2$							
1	1	6	1	$t_2$							
<i>Ps</i> -структура						<i>Ps</i> -структура					
$\bar{P}^*$	$\bar{T}_p$					$\bar{P}^*$	$\bar{T}_p$				
1	7					1	2				

Рис. 9. Примеры альтернативных реляционных отношений, учитывающих циклическость данных

Кортеж  $t_4$  для группы из трех процессов и кортеж  $t_2$  для группы, включающей один процесс, выполняют функцию дополнительного кортежа. На основании данных, которые в них содержатся, можно рассчитать  $T_p$ . Для этого достаточно для каждого экземпляра  $p_p$  провести поиск клона, стартовавшего раньше других, и далее воспользоваться соотношением

$$T_{p=d} = t_{p=d}^m(\bar{C}) - t_{p=d}^n(\bar{C}), \quad (3)$$

где  $m = \max\{t_{p=d}(\bar{C})\}$ ,  $n = \min\{t_{p=d}(\bar{C})\}$ .

Альтернативная конструкция, не использующая дополнительный кортеж, показана на том же рис. 9, б. Значения  $T_p$  заранее рассчитаны и размещены в *Ps*-структуре.

Применение "циклических" отношений сопровождается эффектом компактности. Получим оценку отношения мощностей  $M_p$ - и  $T_r$ -структур:

$$\frac{M(M_p)}{M(T_r)} = \frac{\sum_{p=1}^P p_p}{P \sum_{p=1}^P s_p}, \quad (4)$$

где  $p_p$  — группа процессов;  $s_p$  — порядковый номер группы повторяющихся процессов;  $P$  — число повторяющихся процессов;  $S_p$  — число повторений  $p_p$ .

Очевидно, что с ростом числа повторений растет и отношение (4), следовательно, увеличивается компактность  $M_p$ -структуры.

Исследуем выражение (4), представив его в виде функции  $\frac{M(M_p)}{M(T_r)} = M = f(p, s)$ . Зафиксируем пред-

варительно  $p$ . Тогда  $f(p, s)$  при  $p = \text{const}$  — это гиперболоа, которая достаточно быстро убывает с ростом  $s$ , что свидетельствует о предпочтении использования *Mp*-структуры в сравнении с *Tr*-структурой.

Отметим, что ссылка на квант, идентифицирующий начало процесса (атрибут  $\bar{C}^*$ ), делает ненужным специальным образом фиксировать смещение  $\Delta_p$ . Правда, при этом следует придерживаться соглашения, согласно которому область определения (доменом) атрибута *Chronon* ( $\bar{C}^*$ ) является множество натуральных чисел, что соответствует характеристике введенной ранее последовательности  $\{c_m\}$ .

### Встраивание непериодических циклических данных в циклические структуры

Ситуация, когда отдельные экземпляры пары "тип сущности — тип процесса" не содержат  $pd$ , тогда как большинство других такие данные включают, не является исключительной. Задача совместного размещения *периодических* и *непериодических* повторяющихся данных имеет простое решение.

Идентичность атрибутивного состава *Tr*- и *Mp*-структур позволяет сосредоточить одновременно периодические и непериодические данные в одной

Mp-структура					Ps-структура		
$\bar{E}^*$	$\bar{P}^*$	$\bar{C}^*$	$\bar{r}$		$\bar{P}^*$	$\bar{T}_p$	
1	1	2	1	$t_1$	1	7	$t_1$
1	1	4	1	$t_2$	2	3	$t_2$
1	1	6	1	$t_2$	3	.null	$t_3$
1	2	9	1	$t_4$	Ch-структура		
1	3	7	1	$t_5$	$\bar{C}^*$		
1	3	9	1	$t_6$	1		$t_1$
1	3	15	1	$t_7$	2		$t_2$
					...		
					15		$t_{15}$
					...		

Рис. 10. Использование "нулевых" кортежей для представления неперiodических процессов

единственной структуре. Речь, в частности, идет об *Mp*-структуре. При этом, правда, требуется каким-то образом различать периодические и неперiodические наборы данных. Вариант с использованием дополнительного кортежа *Mp*-отношения не подходит (рис. 9), поскольку неперiodические данные в этом случае могут интерпретироваться как периодические, и наоборот, периодические восприниматься как неперiodические. В то же время использование нулевых (.null) или "пустых" значений, которые может принимать атрибут  $\bar{T}_p$  *Ps*-структуры, является вполне приемлемым решением. Достаточно условиться, что если атрибут  $\bar{T}_p$  какого-либо кортежа *t* *Ps*-структуры равен .null ( $t(\bar{T}_p) = .null$ ), то этот кортеж указывает на неперiodический процесс. Пример, приведенный на рис. 10, наглядно демонстрирует сказанное. Таким кортежем, описывающим и идентифицирующим неперiodический процесс, является кортеж  $t_3$  *Ps*-структуры.

Тем самым, одновременно достигаются две цели: компактное размещение периодических данных и единообразное (в одних и тех же структурах) хранение периодических и неперiodических наборов.

### Привязка прогнозируемых траекторий к реальному времени

Задача привязки к оси реального времени на проверку оказывается не такой тривиальной, как это может показаться. На первый взгляд, достаточно из всех "первых" процессов в парах  $\langle e, p \rangle$  выбрать процесс, стартующий раньше других, и задать время его начала. При более тщательном рассмотрении оказывается, что не исключена вероятность того, что, например, со временем может появиться процесс, начинающийся раньше выбранного. И это только одна из потенциальных нештатных ситуаций. Их все можно исключить, если поставить в соответствие метку времени начала не какому-то отдельному экземпляру процессов, а всему типу  $\langle e, p \rangle$ .

Принимая во внимание существование множества пар  $\langle e, p \rangle$ , в ту часть модели данных, которая обычно отводится под размещение разнообразной метаинформации, логично включить структуру со следующей схемой:

$$R(\text{TimeTuning}) = \{Id\_ob^*, \text{TimeStamp}, Id\_mnc\},$$

где *Id\_ob* — внутренний идентификатор (используемый в конкретной модели данных) *Mp*-структуры; *TimeStamp* — метка времени, синхронизированная с квантами, размещенными в родительской по отношению к *Mp* *Ch*-структуре.

Под синхронизацией здесь подразумевается использование одних и тех же единиц измерения времени, что для квантов, что для меток времени. Если, например, кванты измеряются в днях, то и *TimeStamp* должен принимать значение с точностью до дня. Атрибут *Id\_mnc* в отношении со схемой *R(TimeTuning)* есть не что иное как ссылка на используемую шкалу времени.

Несмотря на то что предложенный вариант привязки носит системный характер, поскольку охватывает все типы пар  $\langle e, p \rangle$ , он пока никак не связан с тем, что можно было бы назвать моделью реального времени базы данных. Построение такой модели — предмет отдельного самостоятельного исследования. В связи с этим уместно сделать ссылку на ряд работ в области темпоральных баз данных [3, 5, 11], авторы которых разрабатывают составные части подобной временной модели. Полученные ими результаты пока охватывают только применяемые временные категории и различные расширения *Entity-Relationship*-технологии в части отражения различных временных аспектов представления сущностей и их взаимодействий в реальной среде.

Указанное обстоятельство делает неполным предложенный вариант привязки, хотя он и включает в себя все наиболее значимые аспекты, в том числе и закрывает вопрос с отражением в моделях данных последнего, не затронутого ранее параметра цикличности — смещения начала первого процесса  $\Delta_p$ .

### Заключение

В работе нашли отражение результаты исследований, касающихся поиска источников, идентификации, моделирования и представления в базах данных периодических данных организационных систем. Показано, что такие данные, как правило, используются в прогнозируемых траекториях и всегда ассоциируются с парой  $\langle \text{сущность}, \text{процесс} \rangle$ . Упомянутая локализация — следствие естественного стремления организационных систем к ритмичному функционированию, которое явно просматривается в фазах планирования.

Присутствие цикличности делает возможным не фиксировать в базах данных всю траекторию, а ограничиться только указанием характеристики цикла. С этой целью в работе построены модели

периодичности и на их основе разработан алгоритм нахождения характеристик периодичности для циклов, состоящих как из одного, так и из нескольких процессов. Предложены моделирующие, в том числе и альтернативные, структуры, позволяющие организовать хранение циклических характеристик в базе данных.

За рамками работы остались несколько вопросов, которые могут продолжить начатое исследование — это встраивание "циклических структур" в подсистему единого модельного времени базы данных (которую еще нужно создать), и подбор моделирующих структур для декомпозируемых и агрегируемых периодических данных с формированием системы соответствующих ограничений. Последняя задача имеет ряд серьезных приложений для функциональной составляющей информационных систем. В частности, это касается нахождения подмножеств зависимых данных, которые должны быть подвержены изменениям вследствие корректировки данных на одном из уровней агрегирования. Функциональные задачи могут охватывать как задачи модификации еще не выполняемых, так и уже частично реализуемых прогнозируемых траекторий.

1. Spaccapietra S., Parent C., Damiani M., Macedo J., Porto F., Vangenot C. A conceptual view on trajectories // *Data and knowledge engineering*. 2008. Vol. 65. P. 126—146.
2. Egidì L., Terenziani P. A modular approach to user-defined symbolic periodicities // *Data and knowledge engineering*. 2008. Vol. 66. P. 163—198.
3. Date C. J., Hugh Darwen, Lorentzos N. A. *Time and relation theory*. Morgan Kaufmann Publishers, 2014. 560 p.
4. Jimenez L. J. REEEM: Reenhancing the entity-relationship model // *Data and knowledge engineering*. 2006. Vol. 58. P. 410—435.
5. Gregerson H., Jensen C. S. Temporal entity-relationship models — a survey // *IEEE Transaction on Knowledge and Data Engineering*. 1999. Vol. 11 (3). P. 464—497.
6. Джексон Г. Проектирование реляционных баз данных для использования с микроЭВМ. М.: Мир, 1991. 252 с.
7. Мейер Д. Теория реляционных баз данных. М.: Мир, 1987. 608 с.
8. Moody D. L. Theoretical and practical issues in evaluating the quality of conceptual models: current state and future directions // *Data and knowledge engineering*. 2005. Vol. 55 (3). P. 243—276.
9. Родионов А. Н. Качество даталогических схем. Принцип компактности моделей данных и его приложения I // *Информатика и системы управления*. 2012. № 1. С. 16—27.
10. Мартин Дж. Организация баз данных в вычислительных системах. М.: Мир, 1980. 663 с.
11. Jensen C. S., Dyreson C. E. (Eds). The consensus glossary of temporal database concepts // *Temporal Databases: Research and Practice*. Springer Verlag. 1998. P. 367—405.

A. N. Rodionov, Leading Researcher, e-mail: ran@newmail.ru,  
Computer Centre of Far-Eastern Branch of RAS

## Periodical Data: Identification, Modeling and Representation in Databases

*There are some tasks concerned with the modeling of such data that differs by the specific properties. This is a data repeated periodically (a cycle data) and a data repeated no periodically. Its representation in databases must capture the separate actions being a parts of entities' lifespan. To that end it is necessary to implement system decision that comprises the temporal and process aspects.*

*In this article we examine the full range of the relevant issues. Thirsts of all the identification characteristics and the cycle attributes are designated. Then some models for finding parameters of cycle processes are constituted. Based on the derived model it introduces and researches the modeling constructions after which compare their quantity. The relations corresponding to modeling structures may utilize to accommodate both cycle and joint data. We also proposal and substantiate the multipurpose structure to store periodical and not periodical data simultaneously.*

*One of the unexpected results appeared to be the fact that the cyclic and acyclic structures, each of which is designed for storage recurrent periodical and recurrent not periodical data, are fully identical. Finally, it articulates how to connect the modeling structures to time axis of database and specifies the advanced fields of study.*

**Keywords:** *periodical and not periodical cycle data, type, instance and clone of entity, multipurpose data structures, temporal databases*

### References

1. Spaccapietra S., Parent C., Damiani M., Macedo J. Porto F., Vangenot C. A conceptual view on trajectories. *Data and knowledge engineering*. 2008, vol. 65, pp. 126—146.
2. Egidì L., Terenziani P. A modular approach to user-defined symbolic periodicities. *Data and knowledge engineering*. 2008, vol. 66, pp. 163—198.
3. Date C. J., Hugh Darwen, Nikos A. Lorentzos. *Time and relation theory*. Morgan Kaufmann Publishers, 2014. 560 p.
4. Jimenez L. J. REEEM: Reenhancing the entity-relationship model. *Data and knowledge engineering*. 2006, vol. 58, pp. 416—435.
5. Gregerson H., Jensen C. S. Temporal entity-relationship models — a survey. *IEEE Transaction on Knowledge and Data Engineering*. 1999, vol. 11 (3), pp. 464—497.
6. Dzhekson G. *Proektirovanie reljacionnyh baz dannyh dlja ispol'zovanija s mikro-JeVM*. М.: Мир, 1991. 252 p.
7. Mejer D. *Teorija reljacionnyh baz dannyh*. М.: Мир, 1987. 608 p.
8. Moody D. L. Theoretical and practical issues in evaluating the quality of conceptual models: current state and future directions. *Data and knowledge engineering*. 2005, vol. 55 (3), pp. 243—276.
9. Rodionov A. N. Kachestvo datalogicheskikh shem. Princip kompaktnosti modelej dannyh i ego prilozhenija I. *Informatika i sistemy upravlenija*. 2012, no. 1, pp. 16—27.
10. Martin Dzh. *Organizacija baz dannyh v vychislitel'nyh sistemah*. М.: Мир, 1980. 663 p.
11. Jensen C. S., Dyreson C. E. (Eds). The consensus glossary of temporal database concepts. *Temporal Databases: Research and Practice*. Springer Verlag. 1998, pp. 367—405.