

УДК 004.09

## ОРИЕНТИРОВАННЫЕ ГРАФЫ В СЛОЖНЫХ ОРГАНИЗАЦИОННО-ТЕХНИЧЕСКИХ ФИСКАЛЬНЫХ СИСТЕМАХ

**Цветков Виктор Яковлевич,**

*профессор, доктор технических наук, заместитель руководителя центра стратегического анализа и развития, Научно-исследовательский и проектно-конструкторский институт информатизации, автоматизации и связи на железнодорожном транспорте (НИИАС), ОАО «НИИАС»  
109029, Россия, Москва, Нижегородская ул., 27 стр. 1  
e-mail: cvj2@mail.ru,*

**Буравцев Алексей Владимирович,**

*заместитель директора Института информационных технологий и автоматизированного проектирования,  
Московский технологический университет (МИРЭА),  
119454, Россия, Москва, Проспект Вернадского, 78,  
e-mail: buravcev@mirea.ru*

*В статье рассмотрены сложные организационно-технические фискальные системы, решающие проблемно ориентированные задачи. Показано место сложной фискальной системы среди других сложных систем. Дано формальное описание модели сложной организационно-технической фискальной системы. Рассмотрено понятие ориентированного ациклического графа. Описаны различные подходы хранения и обработки ориентированных ациклических графов в современных СУБД с использованием языка SQL. Описан подход материализации пути в сложной организационно-технической фискальной системе на примере государственного адресного реестра. Показано, что применение ациклических графов для технологической организации повышает эффективность работы фискальных информационных систем.*

**Ключевые слова:** сложная система, фискальная система, кадастровая система, материализованный путь, ориентированный граф

### ***Введение***

В настоящее время широко применяют различные сложные прикладные системы, которые называют проблемно ориентированными [1]. Такие сложные системы применяют в разных предметных областях, включая области ведения кадастра [2]. Кадастр представляет собой информационную систему содержащую реестр информационных ресурсов или кадастровый реестр [3]. Кадастровые информационные системы служат основой управления и принятия решений [4]. Кадастровый реестр одной из своих основных функций имеет сбор налога с подлежащих налогообложению объектов. Системы, основной функцией которых является сбор налогов называют фискальными системами [5; 6]. Соответственно кадастр делят на фискальный (только сбор налогов, например, земельный налог) и юридический (учет и ведение юридических прав). Юридический кадастр – это в основном земельный кадастр. В данной статье исследуется Федеральная информационная адресная система (ФИАС) [7], относящаяся к фискальным системам.

### ***Материалы и методы***

В качестве материалов использовались результаты существующих исследований в области проблемно ориентированных систем, в области сложных организационных систем, в области систем хранения кадастровой информации и в области кадастровых систем. В качестве методов использовался феноменологический анализ, системный анализ и технологический анализ.

### ***Фискальная система как разновидность сложных систем***

Существуют различные виды сложных систем: сложные технические системы [8], сложные социальные системы [9], сложные организационные системы [10], сложные информационные системы [11], сложные технологические системы [12], сложные системы хранения пространственной инфор-

мации [13], сложные организационно-технические системы [14]. В этом ряду ФИАС относится к двум группам: сложным организационно-техническим системам и сложным системам хранения пространственной информации. Все перечисленные системы решают задачи разных уровней управления.

Управление требует оптимизации налогово-бюджетной политики государства и повышения эффективности налогообложения объектов недвижимости. Это приводит к необходимости построения и эффективного использования фискальной системы как одного из ключевых инструментов налогово-бюджетной политики государства.

При этом необходимо принимать во внимание специфические фискальные [15] и информационные риски [16].

Построение и поддержка в актуальном, полном, достоверном, целостном и непротиворечивом состоянии кадастрового реестра – трудоемкий и финансово затратный процесс [2], требующий, помимо организационных инструментов, активного применения современных алгоритмов обработки данных, представленных в иерархической структуре.

Наличие развитых средств представления и оперирования структурами данных, которые можно представить в виде ориентированного графа, особенно необходимо при конструировании сложных организационно-технических фискальных систем (СОТФС), в которых данные представлены в иерархическом виде. Существуют различные подходы в организации хранения иерархических структур в реляционных БД.

Системы, основной функцией которых является сбор налогов, называют фискальными системами. Результат эволюции фискальных систем привел к необходимости использования для реализации фискальной системы сложной организационно-технической системы (СОТС). Применительно к решению задач налогообложения эта система трансформировалась в сложную организационно-техническую фискальную систему (СОТФС). Системы данного типа являются этапом эволюции сложных систем [17].

Реализация их потенциала требует интеграции многих факторов из разных областей деятельности. Некоторые из этих факторов нематериальны и с трудом поддаются количественной оценке. Например, как эффективно используются информационные ресурсы, насколько эффективно качественное управление, какова адаптивность организации, насколько когнитивной является система управления. Качественные факторы наряду с количественными факторами вносят существенный вклад в эффективное решение сложных задач. Эти качественные оценки часто осуществляют на основе когнитивных методов, что и предопределяет название таких систем как организационно-технические. В настоящее время сложные организационно-технические фискальные системы нуждаются в развитии. Как системы они занимают промежуточное состояние между автоматизированными системами и интеллектуальными системами. По масштабу они относятся к сложным и большим системам. Сложная организационно-техническая фискальная система является разновидностью сложной системы.

В широком смысле под сложной системой понимают множество элементов, находящихся в отношениях и связях друг с другом [18; 19]. В системе эти элементы образуют целостность, единство и обладают эмерджентностью [20]. Содержание элементов системы может изменяться в зависимости от области ее применения и целей.

Необходимо отличать понятие технической системы от понятия формальной системы [21]. Для «системы технического объекта» характерно наличие: технических требований; организационного обеспечения, технической поддержки. Для информационной системы характерно наличие: информационного обеспечения. Для формальной системы характерно наличие информационного языка [22] и лингвистического обеспечения. Проектирование СОТФС происходит по методу «встречных потоков». Оно осуществляется «сверху вниз» на основе декомпозиции [23] и «снизу вверх» на основе агрегации [24].

Разбиение СОТФС на элементы осуществляется в зависимости от критерия делимости системы. Элементы могут объединяться в компоненты системы или в подсистемы. Для формального описания СОТФС (СОТФС) воспользуемся аппаратом теории систем. В качестве описания рассмотрим систему как модель:

$$COTFS = \langle Ps, Pr, Str, E, C, R, G, int, out, \Delta T, Cog \rangle \quad (1)$$

где  $Ps$  – совокупность подсистем системы;  $Pr$  – совокупность частей системы;  $Str$  – структура системы;  $E$  – множество элементов системы;  $C$  – множество связей в системе;  $R$  – множество отношений между элементами, частями и подсистемами;  $int$  – множество входов;  $out$  – множество выходов системы;  $\Delta T$  – интервал жизненного цикла системы;  $Cog$  – когнитивный фактор. Это определение указывает, что система состоит из разнородных частей и имеет структуру. В случае СОТФС структуру можно представить в виде ориентированного ациклического графа.

### **Ориентированные графы**

Во многих областях науки и техники важную роль играет понятие иерархии, иерархической структуры. Для таких областей как теория систем, теория управления, типичным является представление объекта исследования (системы) в виде иерархической структуры подчиненности одних элементов системы другими [10].

Математической моделью иерархических структур являются ориентированные графы. Ориентированным графом (орграфом) называется кортеж  $\langle V, E \rangle$ , где  $V$  – это множество вершин,  $E = \{(v_1, v_2) : v_1, v_2 \in V\}$  – множество дуг. Присутствие во множестве  $E$  вектора  $(v_1, v_2)$  соответствует наличию связи (дуги) между вершинами  $v_1$  и  $v_2$  графа. Тогда иерархией, или иерархической структурой называют ациклический орграф. Орграф называется ациклическим, если в нем нет циклов – последовательностей дуг вида  $(v_1, v_2), (v_2, v_3), \dots, (v_n, v_1)$ , т.е. он не содержит контуров.

Важным частным случаем ориентированных ациклических графов являются деревья. Класс деревьев занимает в теории графов особое положение. С одной стороны, это достаточно просто устроенные графы, и многие задачи, весьма сложные в общей ситуации, для деревьев решаются легко. С другой стороны, деревья часто встречаются в областях, на первый взгляд не имеющих отношения к теории графов. Деревом называется связный граф, не содержащий циклов. Любой граф без циклов называется ациклическим (или лесом). Таким образом, компонентами леса являются деревья.

Если рассматривать структуру кадастровых реестров, таких как КЛАДР и ГАР, являющихся ключевой частью СОТФС ФИАС, то можно констатировать, что она представлена в виде дерева.

### **Структуры ориентированных ациклических графов в SQL базах данных**

Хранение иерархических данных (в виде ориентированных ациклических графов или деревьев) в реляционных структурах задача нетривиальная и вызывает некоторые сложности. В первую очередь, это связано с тем, что реляционные базы не приспособлены к хранению иерархических структур (как, например, XML-файлы), структура реляционных таблиц представляет собой простые списки.

Иерархические данные имеют связь «родитель-наследники», которая не реализована в реляционной структуре. Тем не менее, задача хранения орграфов и деревьев в реляционной базе данных является актуальной при работе в СОТФС, чьи данные моделируются в иерархическом виде. Существуют подходы в организации хранения иерархических структур в реляционных БД [25].

Пусть существуют некие данные, которые можно представить в виде ориентированного ациклического графа или дерева. Рассмотрим основные операции, необходимые для оперирования этими данными в СУБД на языке SQL:

- вставить новый узел;
- удалить узел;
- получить непосредственного предка;
- получить всех предков;
- получить прямых потомков;
- получить всех потомков.

Каждый запрос к реляционной базе данных выполняется за определенное время, поэтому мы стараемся использовать вышеперечисленные операции, используя как можно меньше запросов.

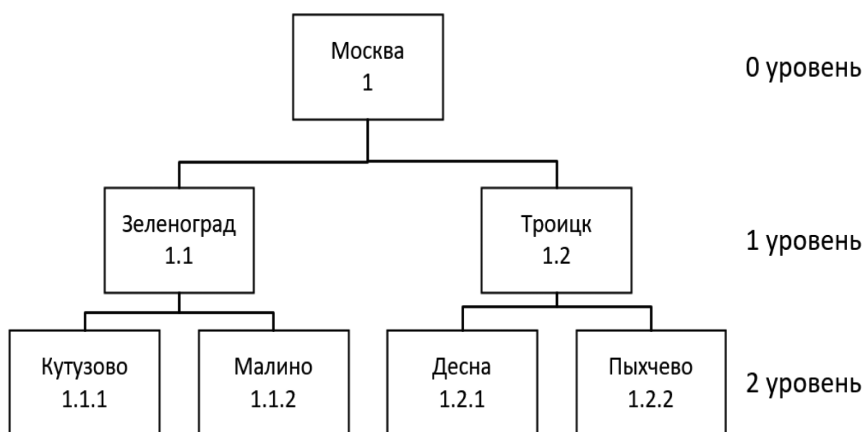
Существует несколько подходов при проектировании баз данных для хранения и обработки иерархических структур:

- материализованный путь (materialized path или path enumeration);
- вложенные множества (nested sets);

- список смежности (adjacency list);
- таблица связей (closure table).

**Материализованный путь (Materialized Path или Path Enumeration)**

Одним из свойств деревьев является то, что есть один и только один путь от корня до каждого узла в дереве. Основная идея алгоритма материализованного пути в сохранении полного пути к узлу от вершины дерева. Модель перечисления путей сохраняет этот путь в виде строки путем конкатенации либо ребер, либо ключей узлов в пути. Поиски выполняются с помощью строковых функций и предикатов для этих строк пути. Существует два метода перечисления путей, перечисления ребер и перечисления узлов. Перечисление узлов является наиболее распространенным явлением, и в базовых операциях строк на обеих моделях мало различий. Тем не менее, модель перечисления границ имеет некоторые полезные числовые свойства. Данные можно представить в виде схемы «материализованный путь» (materialized path) [26] (рисунок 1) применительно к кадастровой системе.



**Рисунок 1 – Древоидная структура, организованная по принципу «материализованный путь»**

Принцип формирования таких путей достаточно прост. Глубина пути – это уровень дерева. Внутри ветки инкрементная нумерация. Другими словами, Зеленоград – первый дочерний элемент Москвы, Троицк – второй дочерний элемент и т.д. Так же возможен вариант, когда путь формируется из идентификаторов узлов. В этом случае, при вставке и обновлении данных нужно проверять, в случае необходимости, путь на отсутствие циклической ссылки (circular reference), когда узел является своим же собственным предком. В случае же инкрементной нумерации узлов внутри ветки данная проверка не нужна.

Идентификаторы узлов в пути можно разделять служебным символом, например, точкой (см. пример ниже), либо конкатенировать значения, но при этом длина символов каждого узла должна быть фиксированной, чтобы потом корректно выделять идентификаторы узлов или ребер в строке при построении запросов. Последний подход позволяет сократить объем строки по сравнению с подходом с разделителем, но несет риски перестроения всех путей в случае изменения длины значения узла или в случае изменения количества уровней.

Данный подход использовался в Классификаторе адресных данных Российской Федерации (КЛАДР) при формировании классификационных кодов адресообразующего объекта и объектов недвижимости. В случае стабильной адресной структуры данная модель вела себя предсказуемо. При добавлении новых уровней, она потребовала изменения длины классификационного кода вследствие добавления новых идентификаторов узлов, после чего потребовалось бы перестроение этого кода у всех объектов. Т.к. данный классификационный код играл роль идентификатора и использовался во множестве смежных систем, то его изменение повлекло бы за собой объемные каскадные изменения во всех этих системах.

Пришлось искать другой путь выхода из сложившейся ситуации, а именно, для КЛАДР несколько уровней государственного адресного реестра объединялись и формировались на одном уровне, например, уровень элементов планировочной структуры и улиц формировался на уровне улиц, что ло-

гически неправильно, но было меньшим из зол. Так как адресная структура пусть не часто, но изменяется, подход, описанный выше, на практике показал свои слабые места. Поэтому лучшим, т.е. более приспособленным к изменениям структуры вариантом является вариант с разделителями и использованием в пути идентификаторов узлов. За гибкость придется платить объемом строки, т.к. с добавлением новых записей размер идентификатора будет расти.

Если же использовать подход с инкрементной нумерацией внутри ветки, то в этом случае нельзя привязываться к материализованному пути как к ключу, т.к. возможны его изменения во времени вследствие изменения количества уровней. Рассмотрим примеры запросов на примере с инкрементной нумерации внутри ветки, как одного из вариантов представления материализованного пути адресного пространства.

Данные будут выглядеть согласно таблицы 1.

**Таблица 1 – Представление данных древовидной структуры**

id	name	path
1	Москва	1
2	Зеленоград	1.1
3	Троицк	1.2
4	Кутузово	1.1.1
5	Малино	1.1.2
6	Десна	1.2.1
7	Пыхчево	1.2.2

Рассмотрим некоторые запросы. Поиск пути к узлу «Кутузово»:

```
SELECT t1.name, t1.path FROM ao_tree t1
CROSS JOIN ao_tree t2
WHERE t2.path like CONCAT( t1.path, '.*')
AND t2.id = 4 ORDER BY t1.path;
```

Результат:

name	path
Москва	1
Зеленоград	1.1

Рассмотрим задачу вычисления уровня вложенности. Для этого достаточно посчитать количество точек (или других разделителей, если используется не точка) в путях. К сожалению, в стандартах SQL нет подходящей функции, но ее можно реализовать косвенным способом, либо написать собственную функцию. Для разделителей, состоящих из одного символа, можно воспользоваться следующим запросом:

```
SELECT LEN(path) – LEN(REPLACE(path, '.', '')) FROM ao_tree
WHERE id=4
```

Если длина разделителя больше одного символа, то можно использовать следующую функцию (на диалекте T-SQL):

```
CREATE FUNCTION dbo.[GetLevel]
(
    @str nvarchar(max),
    @delimiter NVARCHAR(MAX)
)
RETURNS INT
AS
BEGIN
    return (LEN(@str)-LEN(REPLACE(@str,@delimiter,'')))/LEN(@delimiter)
```

END

Рассмотрим задачу выбора ветки узла на примере «Зеленограда»:

```
SELECT name, dbo.GetLevel( path, '.') AS level
FROM ao_tree
WHERE path LIKE '1.1%';
```

Результат:

name	level
Зеленоград	1
Кутузово	2
Малино	2

Обратите внимание, в данном примере мы воспользовались функцией GetLevel, описанной выше. Найдем непосредственный родительский элемент (immediate parent) для узла на примере «Кутузово»:

```
SELECT t2.*
FROM ao_tree t1
CROSS JOIN ao_tree t2
WHERE t1.id=4 AND t2.path=SUBSTRING( t1.path, 1, (LEN(t1.path) - CHARINDEX('.', REVERSE(t1.path))));
```

Результат:

id	name	path
2	Зеленоград	1.1

### Заключение

Использование ориентированных ациклических графов повышает эффективность сложных информационных систем хранения пространственной информации. Оно является эффективным методом контроля и обработки адресной информации в кадастровых системах. Этот подход является важным направлением в совершенствовании фискальных систем. Это обусловлено тем, что адресный реестр является основным составным элементом, формирующим СОТФС. Он влияет на реализацию не только налогообложения, но и всего спектра стратегических задач государства, от создания рыночного оборота недвижимости до формирования доходной налоговой базы. Для таких областей как адресный реестр типичным является представление системы в виде структуры подчиненности одних элементов системы другими. Математической моделью иерархических структур являются ориентированные графы.

Правильно выбранный подход или подходы при проектировании и реализации адресного реестра СОТФС позволит решить задачи эффективного хранения и оперирования иерархическими данными.

### Список литературы

1. Цветков В.Я. Разработка проблемно ориентированных систем управления. – М.: ГКНТ, ВНИЦентр, 1991. – 131 с.
2. Купа Л.В. и др. Ведение кадастра недвижимости на территории с особым правовым режимом государственного природного заказника краевого назначения «Стрижамент» // Применение современных ресурсосберегающих инновационных технологий в АПК. – 2013. – С. 100–103.
3. Маставичене Т.В. Совершенствование технологии ведения адресного реестра для повышения эффективности информационной системы кадастра недвижимости: дис. ... канд. техн. наук: 25.00.26. – М.: МИИГАиК, 2016.
4. Цветков В.Я. Методы поддержки принятия решений в управлении. – М.: Минпромнауки, ВНИЦ, 2001. – 75 с.

5. Балацкий Е.В. Эффективность фискальной политики государства // Проблемы прогнозирования. – 2000. – № 5.
6. Балацкий Е.В. Оценка влияния фискальных инструментов на экономический рост // Проблемы прогнозирования. – 2004. – № 4.
7. Федеральная информационная адресная система. URL: <http://fias.nalog.ru/> (дата обращения: 13.09.2017).
8. Гайкович А.И. Основы теории проектирования сложных технических систем. – СПб.: Моринтех, 2001.
9. Парсонс Т. Социальные системы // Личность. Культура. Общество. – 2003. – Т. 5. – № 1-2. – С. 169–203.
10. Морозов А.А. Ситуационные центры-основа управления организационными системами большой размерности // Математические машины и системы. – 1997. – № 2. – С. 7.
11. Монахов С.В., Савиных В.П., Цветков В.Я. Методология анализа и проектирования сложных информационных систем. – М.: Просвещение, 2005. – 264 с.
12. Гончарова Н.П., Перерва П.Г., Яковлев А.И. Новые технологические системы: качество, потребность, эффективность. – Киев: Наукова думка, 1989.
13. Савиных В.П., Соловьёв И.В., Цветков В.Я. Развитие национальной инфраструктуры пространственных данных на основе развития картографо-геодезического фонда Российской Федерации // Известия высших учебных заведений. Геодезия и аэрофотосъемка. – 2011. – №5. – С. 85–91.
14. Борисов В.В. и др. Компьютерная поддержка сложных организационно-технических систем. – М.: Горячая линия–Телеком, 2002.
15. Пименов Н.А. Фискальные риски в системе налоговой безопасности предприятий и государства // Налоги. – 2010. – № 4. – С. 10–14.
16. Tsvetkov V.Ya. Information Asymmetry as a Risk Factor // European researcher. Series A. – 2014. – Vol. (86). – № 11-1. – Pp. 1937–1943.
17. Корнаков А.Н. Модель сложной организационно-технической системы // Перспективы науки и образования. – 2015. – №2. – С. 44–50.
18. Кудж С.А. Системный подход // Славянский форум. – 2014. – № 1(5). – С. 252–257.
19. Уемов А.И. Системный подход и общая теория систем. – М. Мысль, 1978. – 272 с.
20. Цветков В.Я. Эмерджентизм // Международный журнал прикладных и фундаментальных исследований. – 2017. – № 2-1. – С. 137–138.
21. Кудж С.А. Многоаспектность рассмотрения сложных систем // Перспективы науки и образования. – 2014. – №1. – С. 38–43.
22. Цветков В.Я. Язык информатики // Успехи современного естествознания. – 2014. – №7. – С. 129–133.
23. Решмин С.А., Черноусько Ф.Л. Синтез управления в нелинейной динамической системе на основе декомпозиции // Прикладная математика и механика. – 1998. – Т. 62. – № 1. – С. 121–128.
24. Хрусталёв Е.М. Агрегация данных в OLAP-кубах // Открытые системы. – 2003. – № 5. – С. 33–38.
25. Цикритзис Д., Лоховски Ф. Модели данных: пер. с англ. – М.: Финансы и статистика, 1985.
26. Kuno H.A., Rundensteiner E.A. Augmented inherited multi-index structure for maintenance of materialized path query views // Research Issues in Data Engineering, 1996. Interoperability of Nontraditional Database Systems. Proceedings. Sixth International Workshop on. – IEEE, 1996. – С. 128–137.

### ORIENTED GRAPHS IN COMPLEXS ORGANIZATIONAL AND TECHNICAL FISCAL SYSTEMS

**Tsvetkov Victor Yakovlevich,**

*Professor, Doctor of Technical Sciences,*

*Center for strategic analysis and development, the deputy head,*

*Research and Design Institute of design information, automation and communication on railway transport,*

*27, bldg 1 Nizhegorodskaya Str.,*

*Moscow, Russia,*

*e-mail: cvj2@mail.ru,*

**Buravtsev Alexey Vladimirovich,**

*Deputy Director of the Institute of Information Technologies and Computer-Aided Design, Moscow Technological University (MIREA),  
119454, Vernadsky Prospekt, 78,  
Moscow, Russia,  
e-mail: buravcev@mirea.ru*

*Paper examines complex organizational and technical fiscal systems that solve problem-oriented tasks. The article defines the place of a complex fiscal system among other complex systems. The article provides a formal description of the model of a complex organizational and technical fiscal system. Paper examines the concept of an oriented acyclic graph. The article describes various approaches to storing and processing oriented acyclic graphs in modern DBMS using the SQL language. Paper describes the approach of materializing the way in a complex organizational and technical fiscal system using the example of the state address register. Paper proves that the use of acyclic graphs for a technological organization increases the efficiency of fiscal information systems.*

**Keywords:** complex system, fiscal system, cadastral system, materialized path, oriented graph