

## Модели и методы оперативного формирования документов

Т. Г. ПЕНЬКОВА

*Институт вычислительного моделирования СО РАН, Красноярск, Россия*  
e-mail: Penkova\_t@icm.krasn.ru

Представлено решение задачи оперативного формирования документов, основанное на формальном моделировании информационной структуры и процессов генерации документов. Рассмотрена реализация предложенных моделей и методов на основе интеграции средств управления документами со средствами ведения справочников и OLAP-инструментарием.

*Ключевые слова:* шаблон документа, язык разметки, информационная структура документа, средства формирования документа.

### Введение

Документационное обеспечение управления предполагает формирование различного рода отчетной и организационно-распорядительной документации, что является важным аспектом деятельности любой организации. Подготовка и формирование документированной информации — очень трудоемкий процесс, занимающий большую часть рабочего времени сотрудников. Очевидна необходимость разработки средств, позволяющих автоматизировать процесс производства документов, с использованием информации, накопленной в базах данных.

Разнообразие, сложность форм документов и необходимость изменять их структуру требуют разработки моделей и методов в этой сфере, позволяющих, с одной стороны, автоматизировать разработку и модификацию документов в зависимости от текущих потребностей, а с другой стороны, создать доступные пользователям современные офисные продукты.

Предложена функциональная модель процесса генерации, построенная на этапе проектирования средств оперативного формирования документов. Основное внимание уделено построению информационной структуры документа на базе специализированных шаблонов. Представлены язык разметки и правила описания структуры документа, позволяющие разрабатывать сложные документы. Описаны алгоритм анализа информационной структуры и механизмы оперативного построения запросов с помощью OLAP-моделирования, обеспечивающие сложную логику формирования документов. Рассмотрена реализация предложенных моделей и методов на основе интеграции средств управления документами со средствами ведения справочников и OLAP-инструментарием.

Предложенные модели и методы реализованы в автоматизированной системе поддержки размещения муниципального заказа, что позволило формировать документы на всех стадиях размещения муниципального заказа. Обеспечена возможность динамической настройки параметров документа к меняющимся условиям, не требующая программирования или знания структуры базы данных. Средство генерации характеризуется универсальностью и возможностью применения в различных областях.

## 1. Постановка задачи

Задача автоматизации формирования документов включает следующие подзадачи:

- организацию интерфейса и среды разработки документов, облегчающих подготовку и формирование печатных документов;
- обеспечение возможности создания новых форм документов, построения запросов непосредственно пользователями, не обладающими специальными навыками программирования, знаниями физической структуры базы данных или языка SQL;
- организацию хранения построенных запросов и созданных шаблонов документов с возможностью их многократного применения и использования в качестве источника данных для последующих запросов;
- обеспечение возможности точного позиционирования элементов документа, применение средств оформления и форматирования при разработке шаблона документа;
- поддержку работы пользователя с данными в соответствии с терминологией, принятой в конкретной предметной области.

Процесс автоматизированного формирования документов, как правило, включает два основных этапа: первый — создание визуального макета документа (шаблона), второй — сопоставление элементов шаблона с таблицами базы данных и их заполнение. Способы реализации механизмов разработки шаблона и настройки параметров заполнения созданного шаблона определяют универсальность, гибкость средств генерации и возможность оперативно создавать и модифицировать требуемые документы. В то же время нужно обеспечить возможность работы пользователей, не обладающих специальной подготовкой в сфере информационных технологий. Необходимо сочетать оперативность создания документов с возможностью гибкой подстройки к меняющимся условиям.

Рынок информационных технологий сегодня предлагает огромное количество программных средств, направленных на решение задач документационного обеспечения и ориентированных на различные категории пользователей и уровень технической оснащенности [1–4]. Тем не менее анализ программных средств и существующих подходов к формированию документов позволил выявить ряд проблем и ограничений их применения.

Инструменты класса встроенных средств таких, как, например, Quick Report, Report Builder, ACE Reporter, Rave Reports, предоставляют достаточно большие возможности по формированию документов сложной структуры и представлению информации. Обработка шаблонов выполняется программно [1]. Можно заранее заложить в программу шаблоны формируемых документов, созданные с помощью встроенных средств, и предусмотреть определенные возможности по настройке параметров шаблонов. Однако для создания нового документа или изменения существующих шаблонов требуется написание программного кода и построение запросов на языке SQL.

Специализированные средства, выполненные в виде отдельных приложений, ориентированы в первую очередь на пользователей-непрограммистов, и для разработки макета документа используется графический визуальный редактор. Однако структура шаблонов тяжело воспринимается, настройка параметров заполнения требует знания наименований полей и таблиц базы данных [1, 2]. В некоторых средствах сгенерированные документы не поддаются редактированию и необходим дополнительный этап — конвертирование в другой формат, что требует определенных трудозатрат. Безусловными лидерами специализированных средств считаются Seagate Crystal Reports и комплекс продуктов фирмы BusinessObjects. Внешний генератор BusinessObjects соответствует большинству предъявляемых требований. BusinessObjects имеет возможность настройки на предметную область за счет средств семантической прослойки, обеспечивая работу пользователя в терминах предметной области. Недостатком продукта является сложность интерфейса для неподготовленного пользователя. Перечисленные особенности накладывают ограничения на сферу применимости данных средств.

Частично решить проблему оперативности создания документов и обеспечить доступность инструментария для рядового пользователя позволяет использование приложений Microsoft Office или других подобных продуктов [3, 4]. Преимущество использования средств Microsoft Office (например, текстового редактора Microsoft Word) — в его распространенности и наличии развитых функций оформления документов. Однако слабая структурированность шаблонов накладывают ограничения на структуру, содержание, форму представления информации и затрудняет обработку.

С учетом всех этих недостатков и требований, предъявляемых к средствам оперативной генерации документов, необходима реализация комплексного подхода, простого в использовании, не требующего навыков программирования и позволяющего формировать документы сложной структуры.

## 2. Функциональная модель оперативной генерации документов

Предлагаемое решение базируется на технологиях, объединяющих средства управления документами с системой ведения справочников и OLAP-инструментарием [5].

На основе предложенного подхода построена функциональная модель генерации документов (рис. 1).

Процесс автоматизированного формирования документов разделен на четыре основных этапа:

- 1) создание шаблона документа;
- 2) построение запросов к базе данных;
- 3) настройка параметров заполнения шаблона;
- 4) формирование документа по шаблону.

На этапе разработки шаблона для каждого типа документа пользователем создается визуальный макет. В качестве оболочки разработки шаблона используется текстовый редактор (например, Microsoft Word), и шаблон представляет собой файл с расширением doc. Текстовый редактор предоставляет большие возможности по оформлению и точному позиционированию элементов документа. Немаловажным преимуществом использования средств Microsoft Office можно считать их распространенность и популярность.

При разработке шаблона документ структурируется с помощью специализированной разметки. С ее же помощью определяются роли фрагментов документа в его общей

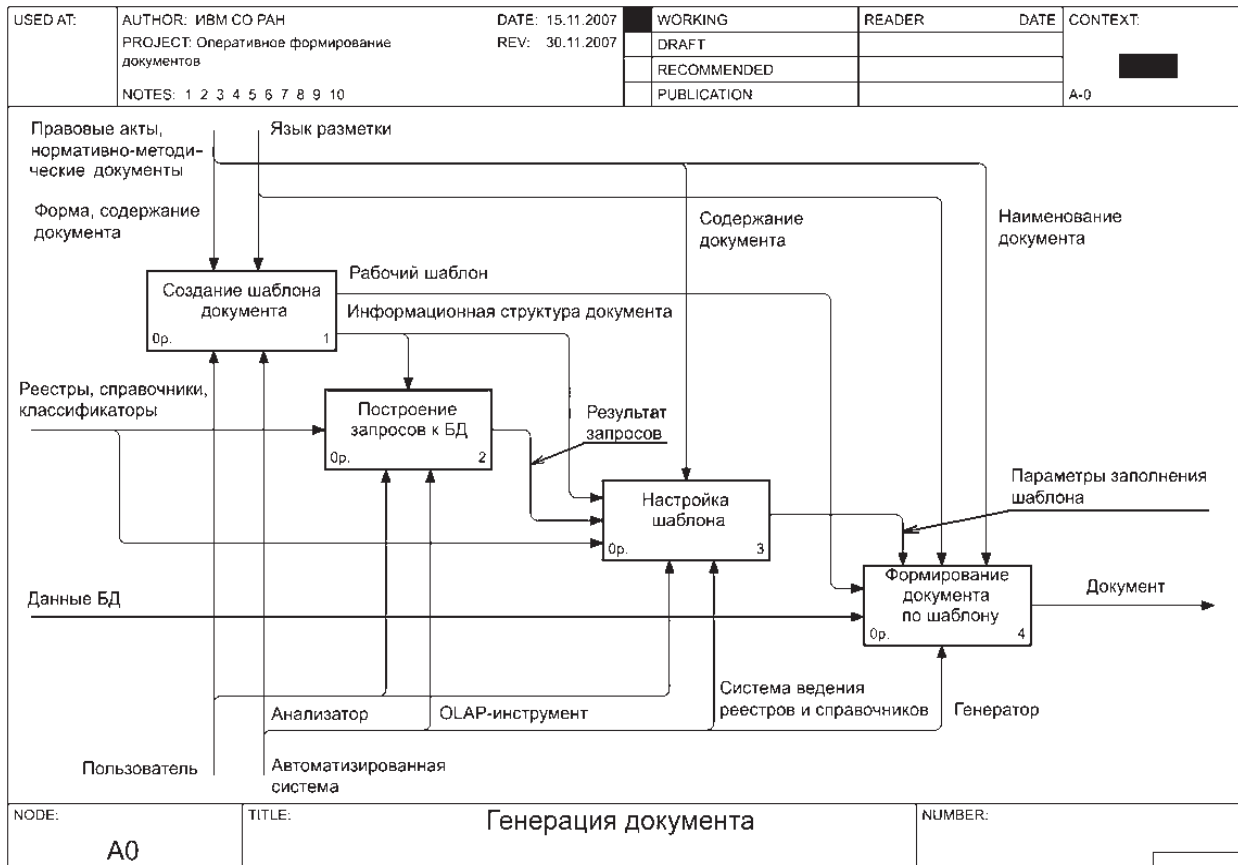


Рис. 1. Функциональная модель процесса генерации документа

структуре, их содержание и визуальное отображение. Специализированная разметка позволяет преодолеть слабую информационную структурированность шаблона и дает возможность описывать документы сложной структуры.

Второй этап процесса генерации — подготовка данных для заполнения созданного шаблона. Для поддержки сложной логики формирования документа, кроме исходных таблиц данных, используются выборки из справочников. Управление выборкой — построение, редактирование, удаление пользовательских запросов, формирование условий — выполняется с помощью механизмов оперативного построения запросов, входящих в состав встроенного OLAP-инструментария. Следствием такой интеграции является возможность строить сложные выражения и запросы с использованием визуального конструктора витрин данных. Анализ информационной структуры документа в соответствии с разработанным алгоритмом позволяет определить необходимость построения запросов и условия их формирования.

На этапе настройки шаблона элементы шаблона сопоставляются с данными исходных таблиц или данными, полученными в результате выполнения запросов. Настройка параметров заполнения шаблона происходит в интерактивном режиме, где каждый элемент информационной структуры документа сопоставляется с полем таблицы базы данных. При этом пользователю не требуется знать служебные наименования таблиц и полей, а также характер связей между таблицами. Все необходимые сведения о структуре таблиц считываются из служебного репозитория встроенной системы ведения реестров и представляются в удобном и наглядном виде.

На последнем этапе формирования документов выполняется автоматическое заполнение шаблона в соответствии с внесенной разметкой, указанными пользователем параметрами и заданными условиями.

### 3. Средства моделирования информационной структуры документа

Слабая структурированность шаблона, созданного в виде электронного документа формата doc, требует применения специальных методов, позволяющих описать структуру любого документа. Для построения модели информационной структуры документа предлагается язык разметки.

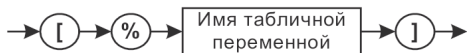
Шаблон содержит два типа информации: статическую, которая остается неизменной в процессе генерации документа, и переменную, которая динамически формируется по результатам выполнения запросов к базе данных. Элементы переменной информации определяются разметкой, которая задается в виде открывающего и закрывающего тегов (меток), с применением специальных символов. Выделены следующие типы элементов переменной информации.

<Переменная> ::=



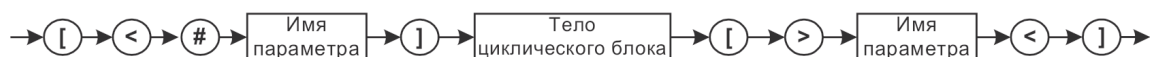
Тип <Переменная> позволяет добавлять в шаблон единицу информации. В процессе генерации документа <Переменная> заменяется значением из поля таблицы базы данных.

<Табличная переменная> ::=



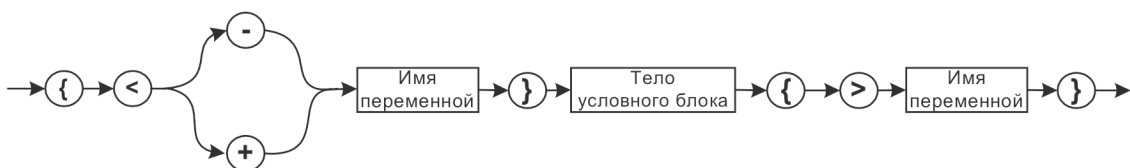
Тип <Табличная переменная> позволяет строить и заполнять таблицу данных. В процессе генерации документа таблица заполняется значениями из базы данных с добавлением строк или столбцов.

<Циклический блок> ::=

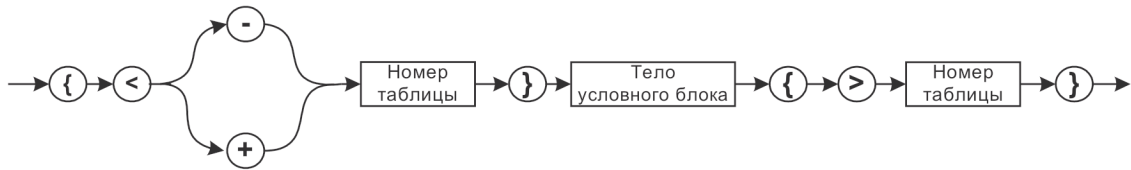


Тип <Циклический блок> обеспечивает многократное повторение фрагмента документа по указанному параметру. Тело циклического блока может содержать любой из типов элементов.

<Условный блок по переменной> ::=



<Условный блок по таблице> ::=



Типы <Условный блок по переменной> и <Условный блок по таблице> обеспечивают визуальное отображение отдельных фрагментов документа соответственно в зависимости от заполнения переменной или таблицы.

Наименование элементов в разметке отражает смысл автоматически вносимой информации. Комплексное применение элементов, циклических и условных блоков позволяет создавать документы различной структуры и обеспечивает реализацию сложной логики формирования документа. Более подробно модель шаблона, язык разметки и правила описания структуры документа рассматриваются в [6].

В результате синтаксического анализа шаблона формируется информационная структура документа в виде дерева элементов переменной информации. Обобщенная схема организации элементов в иерархической структуре представлена на рис. 2.

На рис. 2 содержательная часть документа представлена следующими элементами дерева иерархии:

$R$  — основные реквизиты;

$P_q$  — переменная основных реквизитов, где  $q = \overline{1, Q}$ ,  $Q$  — количество переменных основных реквизитов;

$S$  — основные таблицы;

$T_r$  — основная таблица, где  $r = \overline{1, R}$ ,  $R$  — количество основных таблиц;

$F_s$  — табличная переменная основной таблицы, где  $s = \overline{1, S}$ ,  $S$  — количество табличных переменных основной таблицы;

$V_t$  — цикл, где  $t = \overline{1, T}$ ,  $T$  — количество циклов;

$L$  — реквизиты цикла;

$B_h$  — переменная реквизитов цикла, где  $h = \overline{1, H}$ ,  $H$  — количество переменных реквизитов цикла;

$C$  — таблицы цикла;

$J_f$  — таблица цикла, где  $f = \overline{1, F}$ ,  $F$  — количество таблиц цикла;

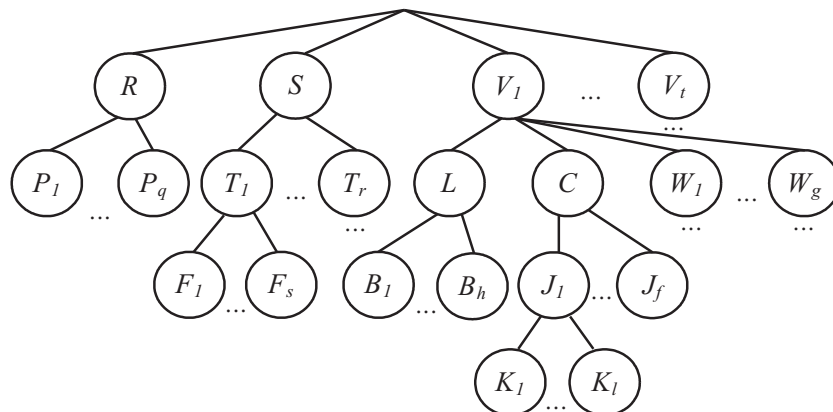


Рис. 2. Обобщенная схема организации элементов в информационной структуре документа

$K_l$  — табличная переменная таблицы цикла, где  $l = \overline{1, L}$ ,  $L$  — количество табличных переменных таблицы цикла;

$W_g$  — вложенный цикл, где  $g = \overline{1, G}$ ,  $G$  — количество вложенных циклов.

Переменная основных реквизитов  $P_q$  характеризует главный параметр, по которому формируется документ. Табличная переменная основной таблицы  $F_s$  характеризует параметр, по которому строится таблица. Цикл  $V_t$  выполняется по параметру, который характеризует главный параметр документа и содержит таблицы цикла, реквизиты цикла и другой цикл — вложенный цикл. Переменная реквизитов цикла  $B_h$  характеризует параметр, по которому выполняется цикл. Табличная переменная таблицы цикла  $K_l$  характеризует параметр, по которому строится таблица цикла. Вложенный цикл  $W_g$  — выполняется по параметру, который характеризует параметр внешнего цикла.

Информационная структура документа в виде дерева иерархии элементов дает возможность связывать переменные элементы документа с полями таблиц и определять необходимость построения пользовательских запросов к базе данных.

#### 4. Алгоритм анализа структуры документа и построение представлений

В процессе генерации документов возникает необходимость построения дополнительных запросов, что вызывается потребностью в новом сочетании существующих данных. Так, для отдельных документов могут потребоваться выборка сведений с определенной фильтрацией, создание расчетных значений или увязка между собой данных, не имеющих прямых ссылок друг на друга. Кроме того, при формировании документов, включающих вложенные элементы (циклы, таблицы и т. д.), используемая технология генерации нуждается в соблюдении правил наследования условий от родительского узла к дочернему узлу: выборка данных для вложенного элемента должна содержать ключевые атрибуты родительского элемента.

Возможность формирования пользователем условий и выражений для поддержки сложной логики формирования документов требует разработки средств управления выборкой данных. Чтобы обеспечить возможность хранения и использования в качестве источника данных результатов ранее выполненных запросов, решено формировать их в виде виртуальных таблиц-представлений (View).

Древовидная информационная структура документа, построенная в результате анализа шаблона, отражает иерархическую информационную зависимость элементов, принадлежность к определенной таблице в базе данных, тем самым позволяя выявить необходимость построения и условия формирования представлений для настройки шаблона. С этой целью выполнена формализация процесса анализа структуры документа в виде алгоритмических схем (рис. 3 и 4).

В алгоритмических схемах, кроме описанных выше обозначений, используются следующие:

$T_k$  — таблица в базе данных, где  $k \in K$ ,  $K$  — количество таблиц в базе;

$a$  — идентификатор главного параметра, по которому формируется документ;

$b$  — идентификатор параметра, по которому строится таблица;

$c$  — идентификатор параметра, по которому выполняется цикл.

На рис. 3 представлен алгоритм анализа элементов дерева с обработкой основных реквизитов и основных таблиц, на рис. 4 — алгоритм анализа цикла. На первом шаге

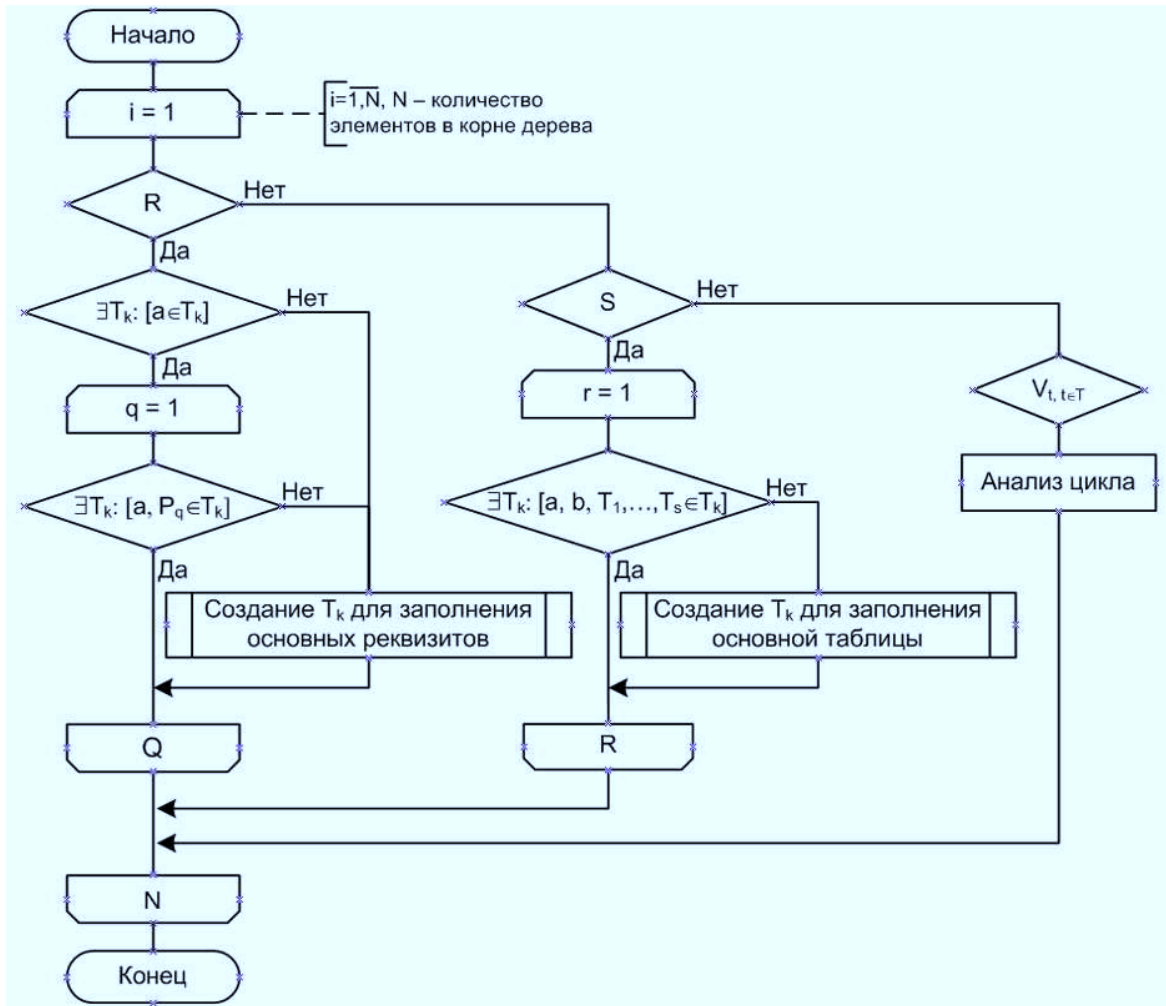


Рис. 3. Алгоритм анализа элементов дерева с обработкой основных реквизитов и основных таблиц

проверяется принадлежность элемента к определенному типу. В зависимости от типа уточняется наличие в базе данных таблицы  $T_k$ , которая содержит идентификационные и дополнительные поля, необходимые для заполнения элемента. При невыполнении условия нужно создать таблицу, содержащую недостающую информацию.

Алгоритм анализа вложенного цикла аналогичен алгоритму анализа цикла. В случае вложенности циклов появляются дополнительные условия для настройки и обработки элементов, входящих во внутренний цикл, — наличие в таблице  $T_k$  идентификатора параметра внешнего цикла, что и отражено в схеме на рис. 4.

Нужная таблица в базе данных создается с помощью механизмов оперативного построения запросов, входящих в состав встроенного OLAP-инструментария, работающего под управлением СУБД Oracle 10g и СУБД семейства Interbase/Firebird. Механизм оперативного построения запросов реализуется с помощью OLAP-моделей, представляющих собой совокупность витрины данных и аналитических операций (формирование показателей, фильтрация, сортировка, выполнение расчетных операций и др.).

Витрина данных формируется при помощи визуального конструктора. В область построения витрины данных из источника данных выбираются таблицы, на основе которых необходимо построить запрос, и между ними устанавливаются связи. Затем



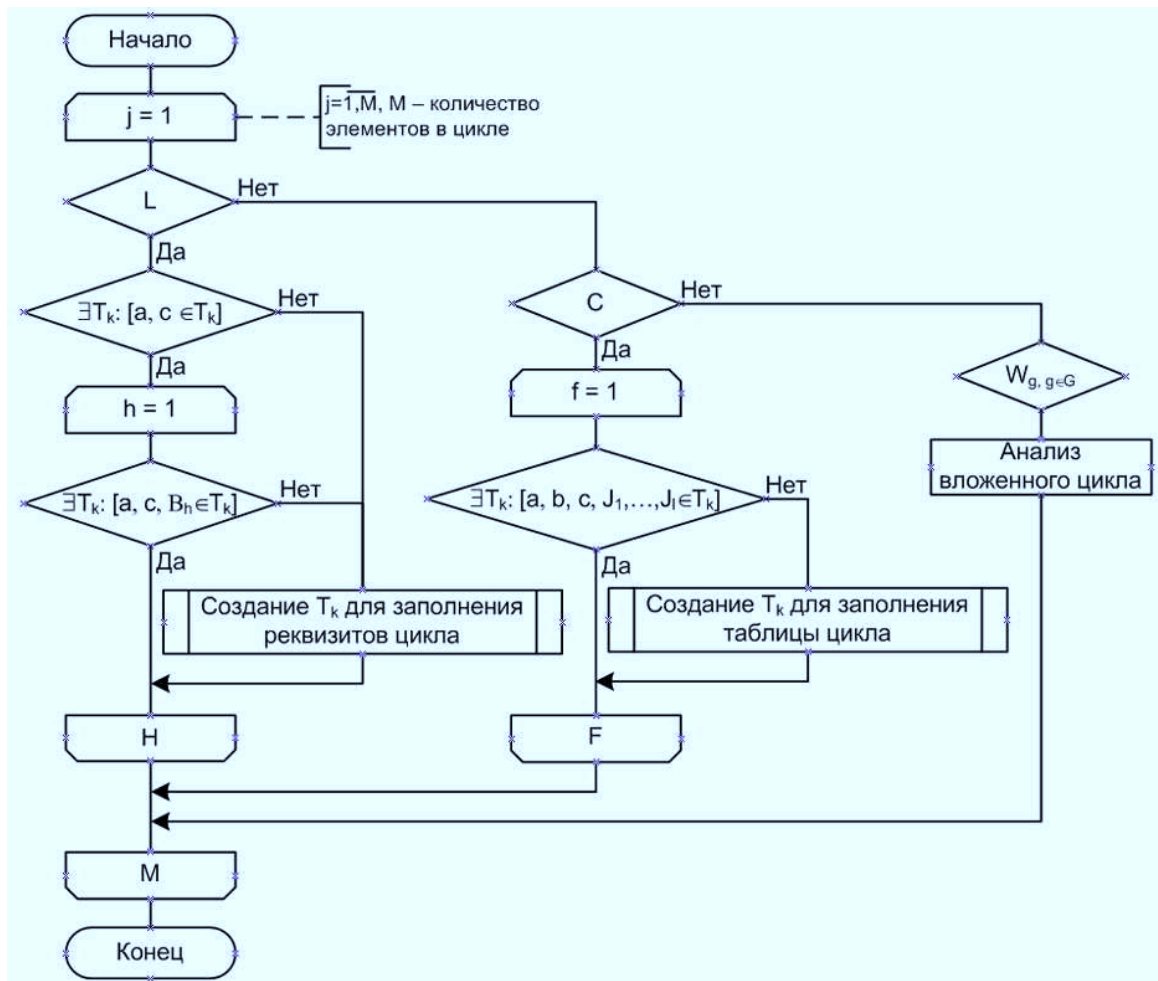


Рис. 4. Алгоритм анализа цикла с обработкой реквизитов цикла и таблиц цикла

из каждой таблицы выбираются нужные поля и задаются условия. После аналитических операций автоматически формируется SQL-запрос, результат выполнения которого представляется пользователю в виде таблицы, сохраняется в базу и может использоваться как источник данных для других моделей. Средства построения запросов учитывают особенности используемых СУБД и производят автоматическую коррекцию запросов.

Интеграция со встроенной системой ведения реестров обеспечивает перенос структур данных между различными схемами или базами данных посредством механизма импорта/экспорта, а также возможность обращения к существующим реестрам и созданным таблицам в терминах предметной области.

## 5. Программная реализация модели генерации документов

Представленные в работе решения, модели, алгоритмы оперативного формирования документов реализованы в автоматизированной системе поддержки размещения муниципального заказа, разработанной специалистами отдела прикладной информатики ИВМ СО РАН. Для реализации подсистемы генерации документов использовались готовые инструментальные средства ведения справочников и OLAP-моделирования [7, 8].

Для организации муниципального заказа разработано более 70 шаблонов документов. На рис. 5 представлен фрагмент шаблона документа “Протокол вскрытия конвертов”.

Пример информационной структуры документа в виде дерева элементов переменной информации представлен на рис. 6.

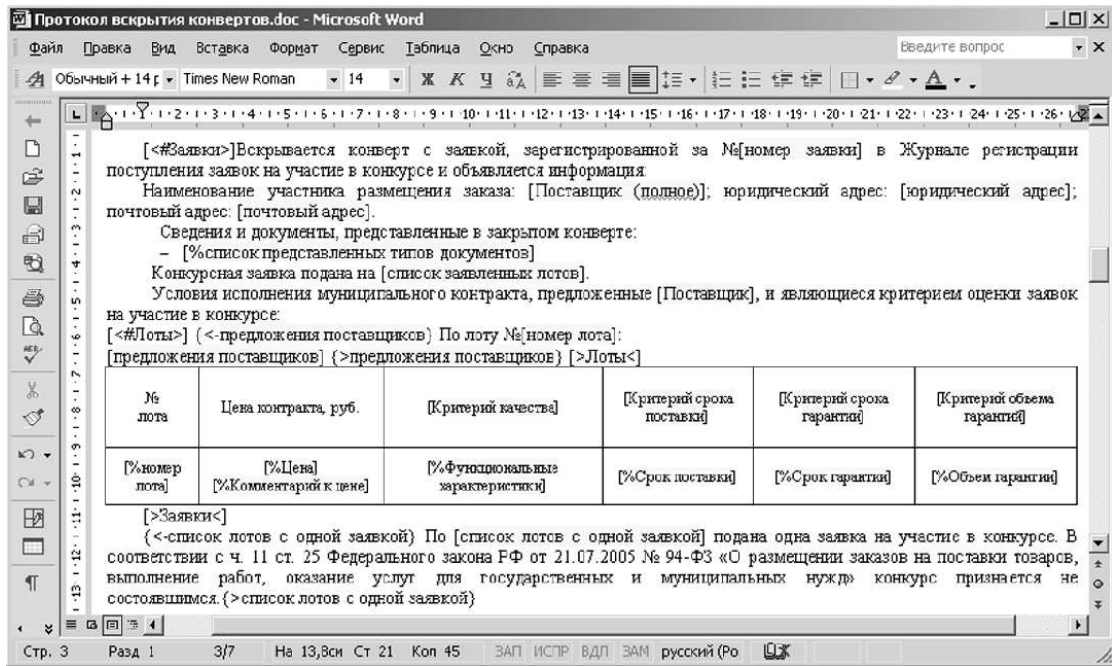


Рис. 5. Фрагмент шаблона документа “Протокол вскрытия конвертов”

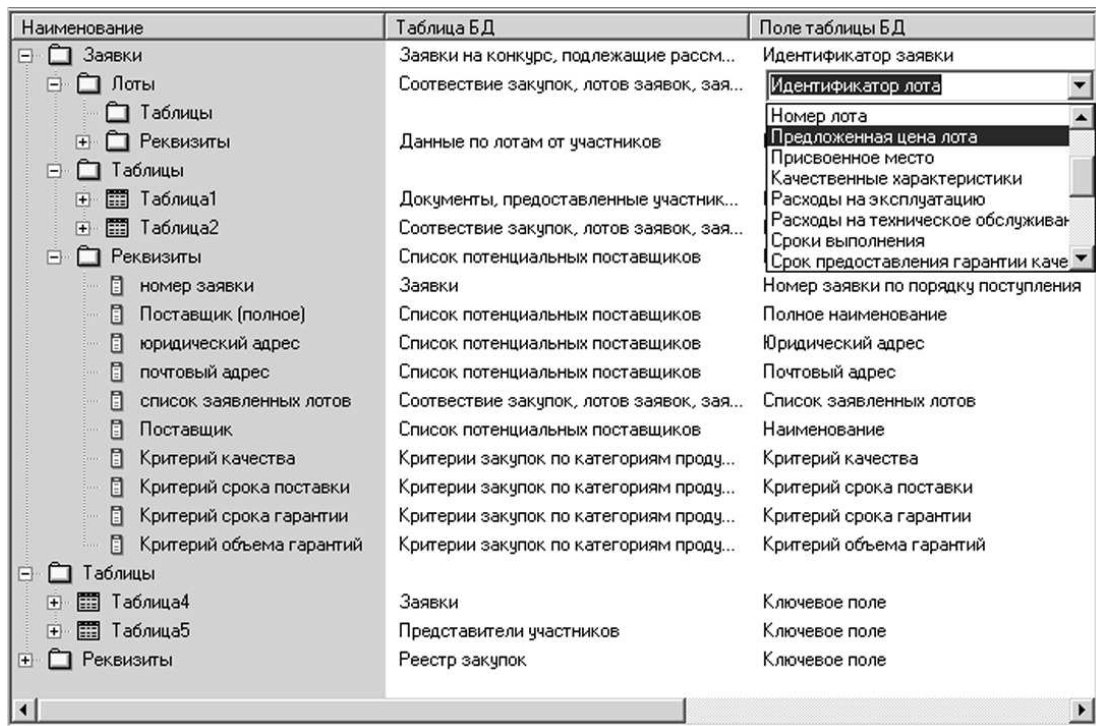


Рис. 6. Дерево элементов информационной структуры документа

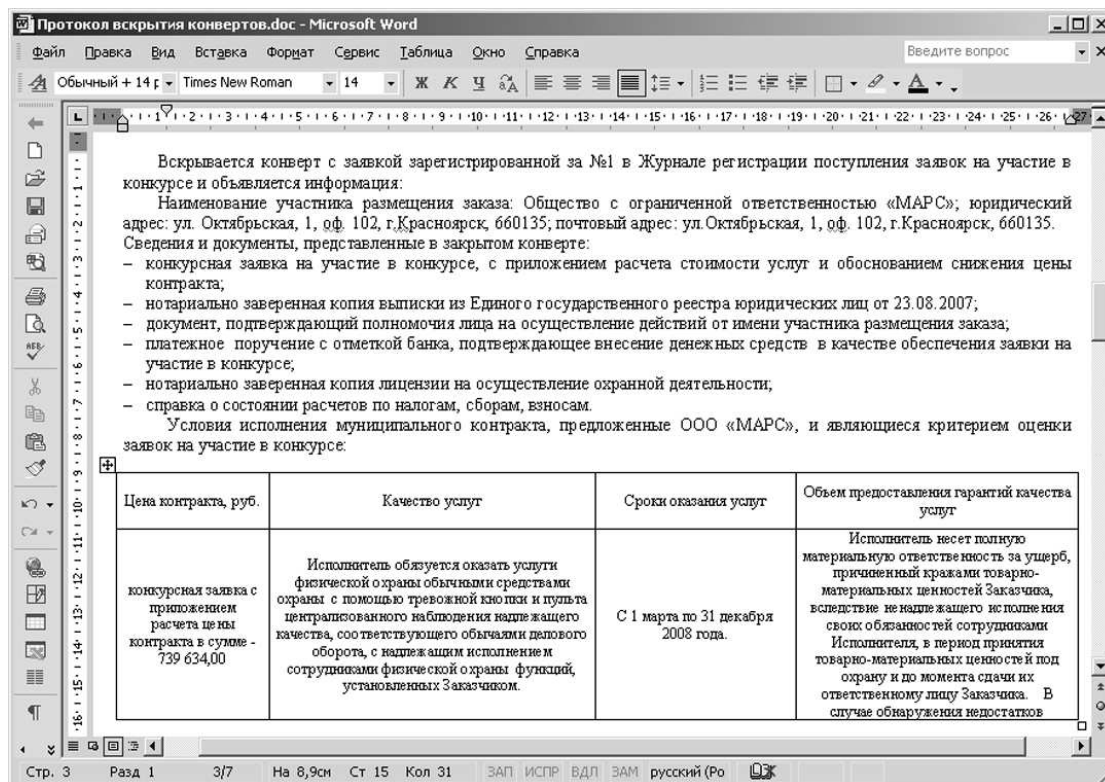


Рис. 7. Фрагмент сформированного документа

Настройка заполнения шаблона (рис. 6) выполняется сопоставлением каждого элемента дерева с полем таблицы базы данных, выбираемым из выпадающего списка. Благодаря репозитарию встроенной системы ведения реестров и справочников, в котором хранится описание таблиц и их свойств (наименования таблиц, оглавлений, полей) в терминах предметной области, пользователь может работать с данными, не зная служебных наименований таблиц и полей, а также характера связей между таблицами. Параметры настройки шаблона сохраняются и используются при многократном формировании документа. В процессе генерации документа в диапазоны шаблона, заданные с помощью специализированной разметки, помещается результат выполнения запросов. Заполнение происходит поэтапно: сначала заполняются основные реквизиты, затем — основные таблицы, циклы и вложенные циклы, на последнем этапе отрабатываются условные блоки.

На рис. 7 представлен фрагмент автоматически сформированного протокола вскрытия конвертов.

С помощью реализованного подхода формируются извещения, сценарии проведения торгов, сводные таблицы об участниках размещения заказа и их предложениях, протоколы заседаний комиссии по рассмотрению, оценки заявок и подведению итогов размещения, уведомления, официальные письма и другие документы, необходимые в процессе организации муниципального заказа.

## Заключение

Предложенный метод структурирования шаблонов с помощью специализированной разметки позволяет создавать сложные документы и реализовать единый подход к пред-

ставлению информационной структуры документа. Информационная структура обеспечивает опциональную настройку элементов шаблона и динамическое заполнение. Организация элементов в структуре документа позволяет определить необходимость и условия формирования пользовательских запросов с помощью предложенного алгоритма анализа. Применение инструментов оперативного построения запросов с использованием OLAP-моделей дает возможность строить сложные выражения, при этом не требуется знание физической структуры базы данных или языка SQL. Встроенная система ведения справочников обеспечивает работу с данными в терминах предметной области.

Практическим результатом стала реализация предложенных моделей, методов и алгоритмов в автоматизированной системе поддержки размещения муниципального заказа. Функционал системы управления документами, объединенный со средствами ведения реестров и OLAP-моделирования, позволяет оперативно формировать полный набор документов, необходимых на всех стадиях размещения муниципального заказа в условиях изменения нормативной базы.

## Список литературы

- [1] Соловьев А.В. Обзор средств генерации отчетов. Концепции и инструментарий // Документооборот. Концепции и инструментарий: Сб. трудов ИСА РАН / Под ред. В.Л. Арлазарова и Н.Е. Емельянова. 2004. С. 174–192.
- [2] Емельянов Н.Е. Особенности систем, создаваемых на основе НИКА-Технологии // Документооборот. Концепции и инструментарий: Сб. трудов ИСА РАН / Под ред. В.Л. Арлазарова и Н.Е. Емельянова. 2004. С. 42–57.
- [3] Смольянинов А. Создание отчетов в средствах разработки Microsoft // Байт. 2005. № 5. <http://www.bytemag.ru/Article.asp?id=3858>
- [4] Ланин В.В. Подсистема управления документами CASE-системы METAS // Математика программных систем: Межвуз. сб. науч. трудов. Пермь, 2006. С. 135–146.
- [5] Жучков Д.В., Пенькова Т.Г. Документоориентированный подход к информационной поддержке размещения муниципального заказа // Матер. 10-й Всероссийской научно-практической конференции ПИР-2007. Красноярск, 2007. Т. 1. С. 77–83.
- [6] Пенькова Т.Г. Информационная модель шаблона для автоматизированного формирования документов // Матер. 10-й Всероссийской научно-практической конференции ПИР-2007. Красноярск, 2007. Т. 1. С. 157–163.
- [7] Жучков Д.В., Коробко А.А. Подход к реализации структуры сложного оглавления справочников в территориально-отраслевых программных комплексах // Информационно-аналитические системы и технологии в здравоохранении и ОМС: Сб. трудов Всероссийской конференции. Красноярск, 2004. С. 239–246.
- [8] Горохова А.В., Ишенин П.П., Никитина М.И. OLAP-средства системы “Аналитик” // Труды Всероссийской конференции “Информационно-аналитические системы и технологии в здравоохранении и ОМС”. Красноярск, 2002. С. 220–228.

*Поступила в редакцию 22 февраля 2008 г.,  
в переработанном виде — 9 июня 2008 г.*