

УДК 004.91

В. В. МИРОНОВ, Г. Р. ШАКИРОВА, В. Э. ЯФАЕВ**ИЕРАРХИЧЕСКАЯ МОДЕЛЬ ПЕРСОНАЛИЗОВАННЫХ ДОКУМЕНТОВ
И ЕЕ XML-РЕАЛИЗАЦИЯ**

Строится модель класса электронных документов в виде иерархии фрагментов, для которых предусмотрена персонализация путем подстановки реквизитов из базы данных. Обсуждается реализация модели на основе технологии XML. Для электронных документов в формате Word предлагается применение XSL-трансформации. *Электронные документы; персонализация; XML; WordML; XSLT*

ВВЕДЕНИЕ

В современном мире доминирующее положение электронных документов не вызывает сомнений. Системы электронного документооборота постепенно вытесняют бумажные носители информации, оставляя за ними, как правило, роль «хранителя подписи» (хотя и подписи тоже могут быть электронными). Главное назначение подобных систем — упростить для пользователей процедуру создания и ведения электронных документов. Все менее актуальным становится некогда распространенный подход использования обобщенных электронных форм документов, в которые пользователь вводит всю совокупность требуемых реквизитов. Сегодня важно не просто предоставить такую форму, а максимально адаптировать ее под конкретного пользователя на основании информации из базы данных.

В этих условиях особую роль играют системы персонализированных электронных документов (ПЭД) — такой позиции придерживаются ведущие производители программного обеспечения. К примеру, программные продукты InfoPath от корпорации Microsoft [4] и LifeCircle [4] от компании Adobe. Это полноценные системы персонализации, предоставляющие простой в обращении инструментарий создания и ведения документов, ориентированных на конкретных пользователей. Однако эти системы работают с электронными документами специфического формата, что вызывает определенные затруднения у конечных пользователей.

В нашей стране наиболее распространенным форматом представления электронных документов был и остается Microsoft Word. Поэтому, чтобы система персонализации была

действительно полезной, она должна поддерживать этот формат или, по меньшей мере, позволять конвертировать документ в формат Microsoft Word.

На сегодняшний день модели и методы построения персонализированных документов проработаны недостаточно. Каждый программный продукт решает задачу персонализации по-своему, что препятствует эффективной обработке электронных документов. В результате существует известная проблема создания унифицированных моделей персонализированных электронных документов.

Нет сомнений, что эта проблема должна быть решена на базе XML-технологий, поскольку в настоящее время XML по праву считается «языком моделирования чего угодно». Такие особенности XML, как открытость, расширяемость и иерархичность способствовали его широкому распространению как формата представления данных. Кроме того, практически все программные технологии предоставляют инструментарий для работы с XML-данными, что существенно повышает привлекательность XML с точки зрения возможности программной обработки. Появление XML-формата для документов Word — так называемого WordML (Wordprocessing Markup Language) — еще один важный аргумент в пользу XML. Документы в таком формате — это, с одной стороны, привычные документы Word, а с другой, — их внутренняя структура представлена в виде XML-разметки.

В статье рассматривается задача формирования персонализированных документов с ориентацией на формат WordML на основе заранее разработанного макета документа и текущего состояния базы реквизитов в формате XML. Она углубляет и детализирует концептуаль-

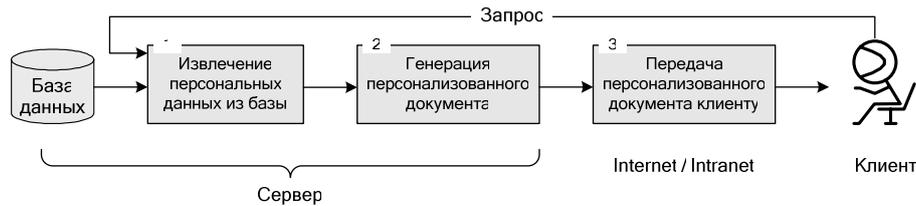


Рис. 1. Общая схема процедуры персонализации электронных документов

ные положения, выдвинутые в работе авторов [1].

1. ОБЩИЕ ПОЛОЖЕНИЯ

Персонализация электронных документов понимается как автоматизированная подстановка пользовательских данных в макет целевого документа. Здесь, очевидно, обязательны обе составляющие. При отсутствии пользовательских данных конечный документ будет представлять собой просто форму, в которую пользователь вынужден вносить данные самостоятельно, а при отсутствии макета формирование документа и вовсе будет невозможно — сами по себе пользовательские данные не образуют целевой документ.

В общем виде процедуру персонализации можно представить следующим образом (рис. 1). Пользователь отправляет на сервер запрос на получение определенного документа. На сервере запускается сценарий, выбирающий соответствующий макет документа и подставляющий в него данные этого пользователя (реквизиты), которые хранятся в специальной серверной базе данных — базе реквизитов. При этом если необходимых реквизитов в базе нет, то пользователь должен отправить соответствующие данные вместе с запросом. После подстановки данных в макет формируется, собственно, целевой документ, который и передается пользователю.

В основе автоматизированного процесса формирования персонализированных документов лежат следующие принципы:

1. Целевой документ, который необходимо сформировать, рассматривается как персонализированный экземпляр некоторого обобщенного класса документов. Класс ПЭД заранее построен в виде иерархии фрагментов: корневой элемент может содержать вложенные внутренние фрагменты, которые, в свою очередь, могут содержать внутренние фрагменты и т.д. Корневой элемент символизирует общий «каркас» документа («пустой» документ), а фрагменты — его информационное наполнение (содержимое). Тогда процедуру персонализации можно представить как

заполнение пустого документа отдельными фрагментами. Порождение конкретного документа как экземпляра класса ПЭД выполняется в соответствии с текущим состоянием базы реквизитов (БР) и внешним запросом, указывающим класс персонализации.

2. Персонализация выполняется в ходе формирования конкретного документа как экземпляра класса в двух направлениях:

а) параметризация фрагментов — в фрагментах предусматривается подстановка определенных реквизитов, значения которых извлекаются из базы реквизитов в соответствии с запросом документа. Модель класса ПЭД должна предусматривать в фрагментах документа ссылки к модели БР, разрешаемые при генерации экземпляра;

б) параметризация структуры — иерархия может предусматривать пропуск того или иного фрагмента, выбор одного из нескольких или, наоборот, копирование одного фрагмента в нескольких экземплярах при порождении экземпляра персонализованного документа. Модель класса ПЭД должна предусматривать управление построением иерархии фрагментов при порождении экземпляра ПЭД в соответствии с состоянием БР и запросом к документу.

3. Процесс автоматизированной генерации ПЭД реализуется на основе XML-технологий, ориентированных на работу с древовидными структурами данных. При таком подходе ПЭД рассматривается как дерево XML-разметки, отдельные фрагменты — как поддеревья, база реквизитов — тоже в виде XML-документа. Процесс формирования ПЭД рассматривается как XSL-трансформация. Класс ПЭД задается XSL-таблицей стилей преобразования XML-базы реквизитов в целевой ПЭД.

2. МОДЕЛЬ ПРОЦЕССА ФОРМИРОВАНИЯ ПЭД

На рис. 2 приведена обобщенная модель процесса формирования персонализованного электронного документа. Исходными данными для него является модель класса персонализированных электронных документов, опи-

сывающая в общем виде некоторый множественный класс персонализированных документов.

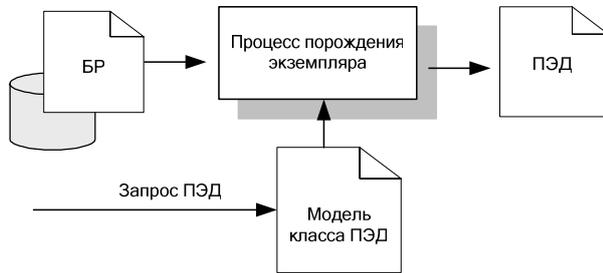


Рис. 2. Общая схема формирования ПЭД

Модель представлена следующими компонентами:

- база реквизитов, хранящая в виде своего текущего состояния реквизиты, на основе значений которых генерируется конкретный документ — экземпляр класса. Под текущим состоянием базы реквизитов понимается совокупность имеющихся данных некоторого пользователя. Текущее состояние реквизитов может меняться, когда пользователь актуализирует информацию — вводит новые данные или меняет уже существующие;
- запрос на персонализацию, уточняющий индивидуальные особенности экземпляра документа, который требуется сгенерировать. Фактически в запросе указывается, какой именно документ нужно сформировать;
- собственно генерацию выполняет процессор преобразования, который в результате обработки исходных данных формирует выходной персонализированный документ.

Рассмотрим более детально составляющие этой модели.

2.1. Модель ПЭД

Модель класса ПЭД (далее просто — модель ПЭД) рассматривается как совокупность фрагментов. Под фрагментом экземпляра документа понимается его часть, имеющая некоторое визуальное представление для пользователя. В модели класса фрагменту соответствует модель фрагмента — настраиваемая часть документа. В модели фрагмента предусмотрена возможность параметрической настройки.

Модель класса ПЭД можно рассматривать на макро- и микроуровне. На макроуровне модель представляет собой конечное множество связанных фрагментов, рассматриваемых как атомарные элементы, на ми-

кроуровне учитывается внутренняя структура каждого фрагмента.

Модель ПЭД на макроуровне

Модель содержит конечное упорядоченное множество элементов $M = \{h\}$. Один из элементов $h \in M$ имеет тип «головной»: $\text{Type}(h) = \text{"head"}$, остальные элементы $f \in M$, $f \neq h$ имеют тип «фрагмент»: $\text{type}(f) = \text{"Frag"}$. $F_M = \{f\}$ — множество всех фрагментов модели. Между парами элементов модели заданы отношения типа «родитель–ребенок»:

- один из фрагментов $f_h \in F_n$ является родителем для головного элемента h : $h = \text{Parent}(f_n)$; f_n называется головным (корневым) фрагментом;
- любой из остальных фрагментов $f \neq f_h$ имеет, по крайней мере, одного родителя: $\forall f', f' \in F_M, f' \neq f_h \exists f'', f'' \in F_M : f'' = \text{Parent}(f')$.

Таким образом, множество M можно рассматривать как множество вершин ориентированного односвязного графа модели документа, дуги которого интерпретируются как отношения типа «родитель–ребенок». Применительно к документу головной элемент рассматривается как точка, с которой начинается обработка модели процессором, головной фрагмент — как укрупненная модель документа в целом, дочерние фрагменты — как модели, на основе которых генерируется внутреннее содержимое некоторых частей документа. Граф модели документа в общем случае не является деревом, хотя совокупность вложенных фрагментов в результирующем документе организована в виде дерева. Это объясняется тем, что одна модель фрагмента, в общем случае, может использоваться в качестве дочерней в нескольких местах документа (иметь несколько родителей).

Модель ПЭД на микроуровне

На этом уровне фрагменты рассматриваются как деревья с упорядоченными нагруженными вершинами, т. е. каждый фрагмент $f \in F_M$ есть некоторое дерево, где $V(f)$ — множество вершин фрагмента f ; $\text{root}(f)$ — корневая вершина; $N(v)$ — нагрузка вершины v , $v \in V(f)$.

Каждая вершина модели фрагмента может быть нагружена значением (value) или ссылкой (ref): $\text{vals}(f)$ — множество вершин фрагмента f , нагруженных значениями; $\text{refs}(f)$ — множество вершин f , нагруженных ссылка-

ми. Если дерево фрагмента задает его структуру (форматирование), то нагрузка вершин задает контент фрагмента (содержимое). В этих условиях ссылки предназначены для настройки фрагмента в ходе порождения персонализированного экземпляра документа.

Ссылка может быть одного из двух типов: реквизитной или фрагментной. Пусть $refs(f)$ – множество ссылок фрагмента f , $j \in refs(f)$ – некоторая ссылка. Тогда предикат $IsReq(j)$ истинный, если j – реквизитная ссылка, а предикат $IsFrag(j)$, если j – фрагментная ссылка.

Реквизитная ссылка j , $IsReq(j)$ интерпретируется как механизм, обеспечивающий при порождении экземпляра документа подстановку значения реквизита из БР в качестве нагрузки соответствующей вершины $Parent(j)$ фрагмента f . Для этого с ней ассоциировано некоторое правило $ReqRule$, позволяющее процессору преобразования извлечь нужное значение реквизита из базы и подставить в фрагмент.

Фрагментная ссылка j , $IsFrag(j)$ интерпретируется как механизм, обеспечивающий при порождении экземпляра документа подстановку в фрагмент $Source(j)$ его дочерних фрагментов, рассматриваемых как внутренние. С каждой ссылкой этого типа связаны один или несколько фрагментов, являющихся дочерними для данного фрагмента. Иными словами, фрагментные ссылки «ссылаются» на фрагменты, реализуя отношения «родитель–ребенок» между фрагментами, введенными при рассмотрении модели на макроуровне.

Из множества $Children(j)$ всех дочерних фрагментов, на которые ссылается ссылка j , при порождении конкретного экземпляра ПЭД выбирается только один. Для этого со ссылкой ассоциировано правило $ChoiceFrag$. В экземпляре ПЭД выбранный фрагмент может копироваться в нескольких экземплярах. Для этого используется правило формирования множества внешних субконтекстов $ChoiceSubset$. Копии экземпляров выбранного фрагмента обрабатываются по-разному, для чего предусмотрены два правила управления внутренним контекстом: $CreaInt$ – правило создания внутреннего контекста, $ModiInt$ – правило модификации внутреннего контекста.

Диаграмма модели ПЭД

Графическое изображение модели ПЭД поясняется на рис. 3, а. Корневой (голов-

ной) элемент модели h изображается овальным символом, соединенным с символом головного фрагмента. Фрагменты изображаются символами-треугольниками со скругленными углами. Кружки внутри символа-фрагмента символизируют вершины дерева фрагмента, при этом корень дерева изображен кружком в верхнем углу символа. Изображения ссылок вынесены за пределы символа-фрагмента; реквизитные ссылки изображаются окружностями, фрагментные ссылки – шестиугольниками. Символ фрагментной ссылки соединяется с несколькими дочерними фрагментами, порядок которых задается метками (номера) или именами в разрывах соединительных линий. Для наглядности на рисунке приведены рассмотренные выше предикаты и формулы, отражающие формальную сторону модели.

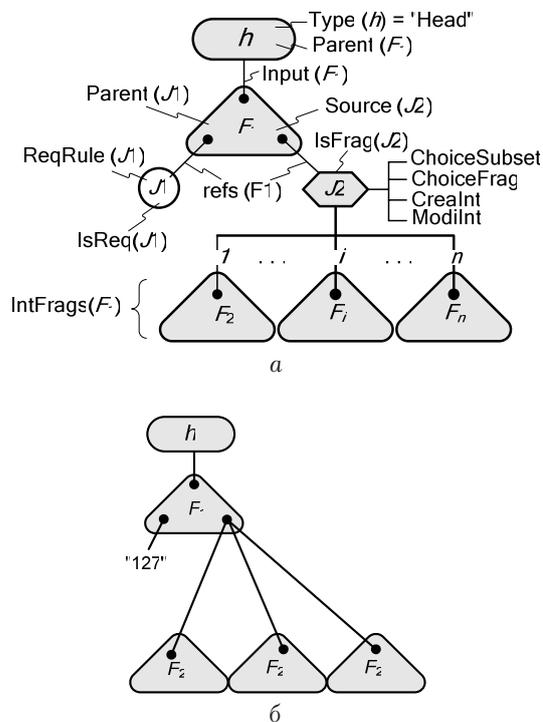


Рис. 3. Графическое изображение модели ПЭД: а – модель; б – вариант экземпляра

На рис. 3, б иллюстрируется вариант сгенерированного экземпляра ПЭД. Здесь в головном фрагменте F_1 ссылка-реквизит J_1 заменена значением «127», а ссылка-фрагмент J_2 заменена соединением с внутренним фрагментами, каковым в данном случае являются три копии фрагмента F_2 .

Частные случаи фрагментных ссылок

В обобщенном виде, представленном выше, фрагментная ссылка обеспечивает тройную функциональность: 1) выбор одного из

дочерних фрагментов; 2) установку контекстов; 3) копирование выбранного фрагмента с установленным контекстом.

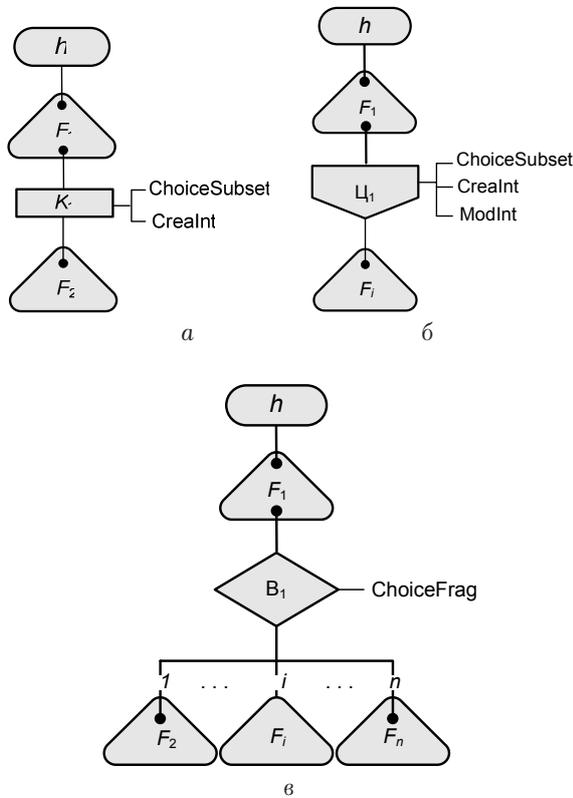


Рис. 4. Частные случаи ссылок: а — контекст; б — выбор; в — цикл

На практике применяются частные случаи фрагментных ссылок (рис. 4):

- ссылка типа «Установка контекста» (а) ссылается на единственный дочерний фрагмент (выбирается всегда), предусматривает правила создания внутреннего и/или внешнего контекстов;
- ссылка типа «Цикл» (б) ссылается на единственный дочерний фрагмент, который при обработке циклических конструкций с различными контекстами, формируемыми с помощью трех правил;
- ссылка типа «Выбор» (в) ссылается на несколько дочерних фрагментов, один из которых выбирается в соответствии с предусмотренным правилом; обработка выбранного фрагмента осуществляется с текущим контекстом.

2.2. Модель базы реквизитов

БР представляется в виде иерархической модели данных — в виде дерева типов реквизитов. Каждый тип реквизита подразумевает множество экземпляров (значений), таким

образом, собственно база реквизитов представляет собой множество экземпляров деревьев, нагруженных значениями реквизитов.

Внешний контекст, учитываемый при порождении экземпляров ПЭД, представляет собой указатель, ссылающийся на одну из вершин некоторого экземпляра дерева реквизитов, при этом при обработке фрагмента с данным контекстом доступны значения реквизитов, соответствующие поддереву, на корень которого указывает контекст.

Для извлечения значений реквизита применяется двухфакторный метод, когда однозначность результата определяется, во-первых, наличием правила, сформулированного в терминах модели типов, во-вторых, наличием некоторого контекста, который уточняет конкретные значения в рамках типа. Например, правилу «извлечь атрибут "фамилия" сотрудника организации» вне контекста соответствует множество значений фамилий различных сотрудников, если при этом текущий контекст соответствует сотруднику с кодом, скажем, «127», то результат сужается до единственного значения, скажем, «Иванов».

2.3. Модель процессора преобразования

Модель процессора поясняет, как производится обработка модели класса ПЭД при генерации экземпляра ПЭД. В основе обработки лежат следующие принципы:

- процесс порождения экземпляра рассматривается как процесс сборки ПЭД из отдельных фрагментов, модели которых содержатся в общей модели ПЭД. Процесс управляется правилами, ассоциированными с реквизитными и фрагментными ссылками в той же модели;
- обработка фрагмента происходит в буфере результата, куда процессор помещает обрабатываемые фрагменты, извлеченные из модели ПЭД. Здесь процессор настраивает фрагмент: реквизитные ссылки он заменяет значениями реквизитов, а фрагментные ссылки — соответствующими внутренними фрагментами;
- рекурсивный характер обработки: чтобы обработать некоторый фрагмент, нужно выполнить подстановку в него внутренних фрагментов, которые предварительно нужно обработать аналогичным образом. У внутренних фрагментов могут быть свои внутренние фрагменты, которые требуют аналогичной обработки;
- внешний контекст — текущее поддерево в иерархической базе реквизитов — зна-

чально задается внешним запросом на персонализацию. В дальнейшем при обработке фрагментных ссылок происходит управление внешним контекстом: в нем формируется субконтекст в виде поддерева текущего контекста, внутренние фрагменты обрабатываются в этом субконтексте;

- внутренний контекст — совокупность переменных, значения которых устанавливаются и изменяются в ходе сеанса обработки, — также управляется в ходе обработки фрагментных ссылок. Обычно он используется при циклической обработке фрагмента с тем, чтобы на текущем цикле учитывать особенности предыдущих циклов. Подстановка реквизитов также может выполняться на основе значений контекста.

Алгоритм преобразования

На верхнем уровне алгоритм преобразования включает последовательные вызовы двух процедур:

```
let (p0, s0) = ProcessHead (Q)
ProcessFrag (p0, s0).
```

Здесь ProcessHead — «обработать головной фрагмент» — начальная процедура, выполняющая подготовительные функции; ProcessFrag — «обработать фрагмент» — основная рекурсивная процедура, выполняющая сборку ПЭД. Процедура ProcessHead в соответствии с внешним запросом Q возвращает в качестве результата указатель p₀ на головной фрагмент модели ПЭД и начальный контекст s₀, с которых нужно начать обработку. Здесь контекст s₀ состоит из двух компонент: внешнего контекста s_E, задающего текущее поддерево в иерархической базе реквизитов, и внутреннего контекста s_I, задающего множество переменных состояния сеанса обработки.

```

proc ProcessFrag (p, s) returns (f)
  let f = ExtractFrag(p)
  for-each j, j ∈ Refs(f)
    if IsReq(j) then InsertReq(j, s)
    else let f* = ChoiceFrag (j, s)
         let si* = CreaInt(j, s)
         for-each sE*, sE* ∈ ChoiceSubset(j, s)
           let f** = ProcessFrag(f*, s*)
           InsertFrag(j, f**)
         let sI* = ModInt(j, s)

```

Рис. 5. Псевдокод процедуры «Обработать фрагмент»

Процедура ProcessFrag («обработать фрагмент») в виде псевдокода представлена на рис. 5. В ходе выполнения процедура обращается к встроенным функциям нижнего

уровня, выполняющим логически самостоятельные задачи манипулирования данными (табл. 1).

Ниже даны пояснения к строкам псевдокода процедуры.

1. Заголовок процедуры ProcessFrag. В качестве входных параметров передаются указатель p на обрабатываемый фрагмент в модели и текущий контекст s. В качестве результата процедура возвращает указатель f на подготовленный фрагмент в буфере обработки.

2. С помощью функции ExtractFrag извлекается модель обрабатываемого фрагмента и помещается в буфер обработки (строка 2).

3. Определяется множество Refs(s) ссылок обрабатываемого фрагмента f и циклически выполняется обработка каждой из них (строки 3–10). Здесь j — текущая обрабатываемая ссылка.

4. Если ссылка реквизитная, то с помощью функции InsertReq(строка 4) извлекается значение реквизита и подставляется в соответствующую вершину фрагмента.

5. В противном случае производится обработка ссылки как фрагментной (строки 5–10).

6. С помощью функции ChoiceFrag выбирается единственный фрагмент f* из всех дочерних для ссылки j, который будет обрабатываться в качестве внутреннего в данном экземпляре ПЭД.

7. С помощью функции CreaInt формируется внутренний контекст s_i* для обработки внутренних фрагментов.

8. С помощью функции ChoiceSubset формируется множество внешних субконтекстов и для каждого из них циклически обрабатывается выбранный внутренний фрагмент f* (строки 7–10). Здесь s_E* — текущий внешний субконтекст обработки.

9. С помощью рекурсивного вызова процедуры ProcessFrag обрабатывается внутренний фрагмент f* в контексте s*. Этот контекст включает в качестве компонент внешний s_E* и внутренний s_i* контексты.

10. С помощью функции InsertFrag обработанная копия внутреннего фрагмента f* вставляется в обрабатываемый фрагмент f в качестве внутреннего.

11. С помощью функции ModInt модифицируется внутренний контекст для следующего цикла обработки.

Функции нижнего уровня процессора преобразования

Имя (название)	Параметры	Содержание
ExtractFrag (извлечь фрагмент)	p — указатель на фрагмент; f^* — указатель на фрагмент в буфере	Извлекается модель фрагмента, на который указывает p , и помещается в буфер для обработки. В качестве результата возвращается указатель f^* на фрагмент в буфере
ChoiceFrag (выбрать фрагмент)	j — фрагментная ссылка s — контекст; f^* — указатель на выбран- ный фрагмент	Для ссылки j извлекается правило $fRules = FragRule(j)$, которое применяется к базе реквизитов в контексте s , в результате чего определяется порядковый номер одного из фрагментов, выбранного из множества внутренних $IntFrag(j)$. В качестве результата f^* возвращает указатель на выбранную модель внутреннего фрагмента
ChoiceSubset (выбрать контекстное подмноже- ство)	j — фрагментная ссылка; s — контекст; S^* — множество внешних контекстов	Для ссылки j извлекается правило $sRule = SubsetRule(j)$, которое применяется к базе реквизитов в контексте s , в результате чего формируется множество внешних субконтекстов $S^* = \{s^*\}$, где каждый субконтекст s^* является поддеревом контекста. Множество S^* возвращается в качестве результата
InsertReq (подставить реквизиты)	j — реквизитная ссылка; s — контекст	Извлекается правило $ReqRule(j)$, соответствующее ссылке j , которое применяется к базе реквизитов в контексте s , в результате чего определяется значение реквизита $req(j, s)$. Это значение поставляется во внутреннюю вершину $Source(j)$ фрагмента $Parent(j)$
InsertFrag (подставить фрагмент)	j — фрагментная ссылка; f — обработанный фраг- мент	Обработанный в буфере фрагмент f прикрепляется в качестве под- дерева к внутренней вершине $Source(j)$ обрабатываемого фрагмента $Parent(j)$
CreaInt (создать внутренний контекст)	j — фрагментная ссылка; s — контекст; s_I^* — внутренний контекст	Извлекается правило $CreaInt(j)$ и в результате его применения в кон- тексте s создается внутренний контекст s_I^*
ModiInt (изменить внутренний контекст)	j — фрагментная ссылка; s — контекст; s_I^* — внутренний контекст	Извлекается правило $ModiInt(j)$ и в результате его применения в кон- тексте s^* создается новая версия внутреннего контекста s_I^*

2.4. Пример модели ПЭД

На рис. 6, а приведен макет электронного документа, содержащего сведения о поставках товаров некоторым поставщиком. Детальное описание компонентов этого макета приведено в табл. 2.

Структуру документа можно представить как совокупность четырех фрагментов (Φ_0, \dots, Φ_3). Фрагмент Φ_0 задает общую структуру документа с двумя полями для ввода данных (номер и название поставщика). Этот фрагмент всегда присутствует в генерируемом документе независимо от значений соответствующих реквизитов. Фрагменты Φ_1 и Φ_2 являются альтернативными — их появление в конечном документе определяется значениями реквизитов. Если для заданного поставщика не определен список поставляемых товаров, то в документ включается фрагмент Φ_1 , в противном случае — фрагмент Φ_2 . Последний фрагмент, в свою очередь, содержит внутренний фрагмент Φ_3 .

Логика построения документа по данному макету описывается моделью ПЭД, приведенной на рис. 6, б. На вход модели подает-

ся параметр «Поставщик.Код_п», в соответствии с которым формируется вся последующая структура. Помимо фрагментов и реквизитных ссылок, в ней предусмотрены две фрагментных ссылки. Ссылка « V_1 » типа «Выбор» задает проверку наличия поставляемых товаров у заданного поставщика и формирует фрагмент Φ_1 или Φ_2 . Ссылка « Π_1 » типа «Цикл» для каждого поставляемого товара поставщика формирует экземпляр фрагмента Φ_3 .

Сопоставление между элементами модели реквизитов и элементами модели ПЭД организовано с помощью ссылок, определяющих точки подстановки реквизитов в соответствующих фрагментах модели ПЭД. Детальное описание управления контекстом с помощью механизма ссылок приведено в табл. 3.

Процесс генерации экземпляров ПЭД, соответствующих примеру модели, был протестирован «вручную» путем мысленного выполнения алгоритма преобразования, рассмотренного выше, для различных вариантов состояния БР. Дальнейшее погружение его работоспособности произведено путем «погружения» модели ПЭД и алгоритма преобразо-

Таблица 2

Спецификация модели ПЭД (см. рис. 6)

Обозначение	Содержание	Пояснение
H_1	Головной элемент, соответствующий модели в целом	Предусмотрен входной параметр «Поставщик.Код-П», идентифицирующий поставщика, для которого генерируется целевой документ
C_0	Фрагментная ссылка «Контекст», ссылается на фрагмент Φ_0	Задаёт начальный контекст генерации документа, соответствующий поставщику, заданному входным параметром
Φ_0	Головной фрагмент	Заголовок документа с двумя полями Π_1 и Π_2 и одним из внутренних фрагментов Φ_1 или Φ_2
Π_1	Реквизитная ссылка	Ссылается на реквизит «Код-П» (код поставщика) элемента «Поставщик» в БР
Π_2	то же	Ссылается на реквизит «Назв-П» (название поставщика) элемента «Поставщик» в БР
V_1	Фрагментная ссылка «Выбор», ссылается на альтернативные фрагменты Φ_1 и Φ_2	Проверяет наличие дочерних элементов «Поставленный-товар» у элемента «Поставщик» в БР
Φ_1	Внутренний фрагмент	Альтернативный фрагмент для случая, когда у поставщика не выявлено поставленных товаров. Содержит соответствующее уведомление
Φ_2	То же	Альтернативный фрагмент для случая, когда у поставщика имеются поставленные товары. Содержит заголовок таблицы поставленных товаров и множественный внутренний фрагмент Φ_3
Π_1	Фрагментная ссылка «Цикл», ссылается на фрагмент Φ_3	Формирует упорядоченное множество элементов «Поставленный-товар», дочерних для элементов «Поставщик» БР, и для каждого из них копирует фрагмент Φ_3 . Формирует внутреннюю переменную n с начальным значением 1, которая инкрементируется после каждого цикла
Φ_3	Внутренний фрагмент	Строка таблицы, соответствующая поставленному поставщиком товару. Содержит поля Π_3 , Π_4 и Π_5
Π_3	Реквизитная ссылка	Ссылается на внутреннюю переменную n (номер по порядку)
Π_4	то же	Ссылается на реквизиты «Назв-Т» (название товара) элемента «Поставленный-товар» в БР
Π_5	то же	Ссылается на реквизиты «Кол-во» (количество товара) элемента «Поставленный-товар» в БР

Таблица 3

Управление контекстом в процессе генерации экземпляра ПЭД

Элемент модели ПЭД	Контекст	Ссылка	Адресуемый объект
M_1	Внешний: все дерево БР	–	–
C_0	Внешний: поддерево БР, корень которого соответствует поставщику с кодом, заданным во входном параметре	Π_1 Π_2	Атрибут «Код-П» контекстного элемента «Поставщик» Значение атрибута «Назв-П» контекстного элемента «Поставщик»
V_1	Контекст не изменился	–	–
Π_1	Внешний: на каждом цикле последовательно элементы «Поставленный-товар», дочерние для элемента «Поставщик» из предшествующего контекста. Внутренний: на каждом цикле переменная n имеет значения 1, 2, 3, ... и т.д.	Π_3 Π_4 Π_5	Значение внутренней переменной n Значение атрибута «Назв-Т» контекстного элемента «Поставленный-товар» Значение атрибута «Кол-во» контекстного элемента «Поставленный-товар»

вания в среду XML и отработки информационной технологии, пригодной для практического использования.

Подстановка значений в поля ввода фрагментов выполняется в соответствии с моделью реквизитов, приведенной на рис. 6, в.

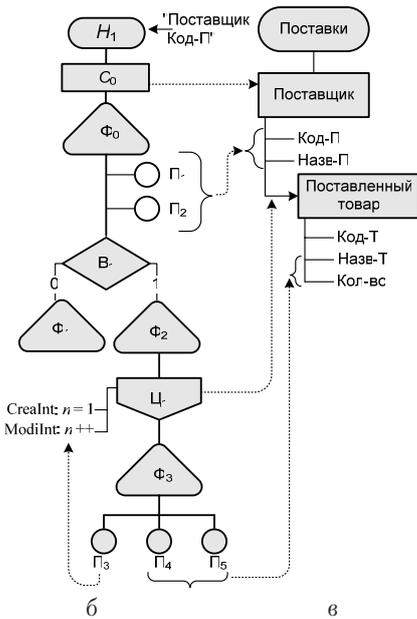
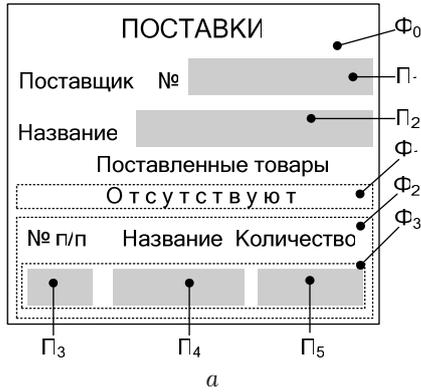


Рис. 6. Пример формирования документа с тремя уровнями вложенности фрагментов: а – макет ПЭД; б – модель класса ПЭД; в – модель базы реквизитов

Корневой элемент модели – «Поставщик» – содержит два атрибута «Код-П» и «Назв-П», значения которых подставляются

во фрагмент Φ_0 . Элемент «Поставщик» содержит множественный необязательный элемент «Поставленный товар», атрибуты которого – «Код-Т», «Назв_Т» и «Кол-во» – подставляются во фрагмент Φ_3 .

3. РЕАЛИЗАЦИЯ МОДЕЛИ ПЭД В СРЕДЕ XML

Рассмотренная выше модель класса ПЭД имеет концептуальный характер в том смысле, что не привязана к конкретной среде реализации и, в принципе, может быть реализована на различных инструментальных платформах. Однако наиболее близкой для нее является платформа XML [2, 3]. Рассмотрим основные принципы и приемы формирования ПЭД в среде XML.

Иерархическая модель БР как нельзя лучше соответствует древовидной организации XML-документов. База реквизитов представляется (реально или виртуально) как XML-документ: структура задается с помощью XML-разметки, значения реквизитов – в виде текстового содержимого элементов (элементоцентричная организация) или в виде значений атрибутов (атрибутоцентричная организация). Например, рассмотренную ранее модель базы реквизитов (см. рис. 6, в) можно представить в форме XML-разметки. Обобщенная структура такой разметки приведена на рис. 7.

Группе реквизитов «Поставщик» соответствует одноименный XML-элемент, являющийся корневым элементом XML-документа. Этот элемент вложен в «корень» XML-дерева – элемент «Поставщики». Реквизиты представлены в элементоцентричной манере, поэтому «Код-Т» и «Код-П» являются XML-элементами, вложенными в элемент



```
<?xml version="1.0" encoding="windows-1251"?>
<Поставки>
  <Поставщик>
    <Код-Г>e1</Код-Г>
    <Назв-Г>Смит</Назв-Г>
    <Поставленный-товар>
      <Код-Т>e1</Код-Т>
      <Назв-Т>Гайка</Назв-Т>
      <Кол-во>100</Кол-во>
    </Поставленный-товар>
    <Поставленный-товар>
      <Код-Т>e2</Код-Т>
      <Назв-Т>Шайба</Назв-Т>
      <Кол-во>200</Кол-во>
    </Поставленный-товар>
  </Поставщик>
  <Поставщик>
    <Код-Г>e2</Код-Г>
    <Назв-Г>Блейк</Назв-Г>
  </Поставщик>
</Поставки>
```

Рис. 7. Пример XML-модели БР: а – иерархическая модель; б – пример реализации XML-разметки

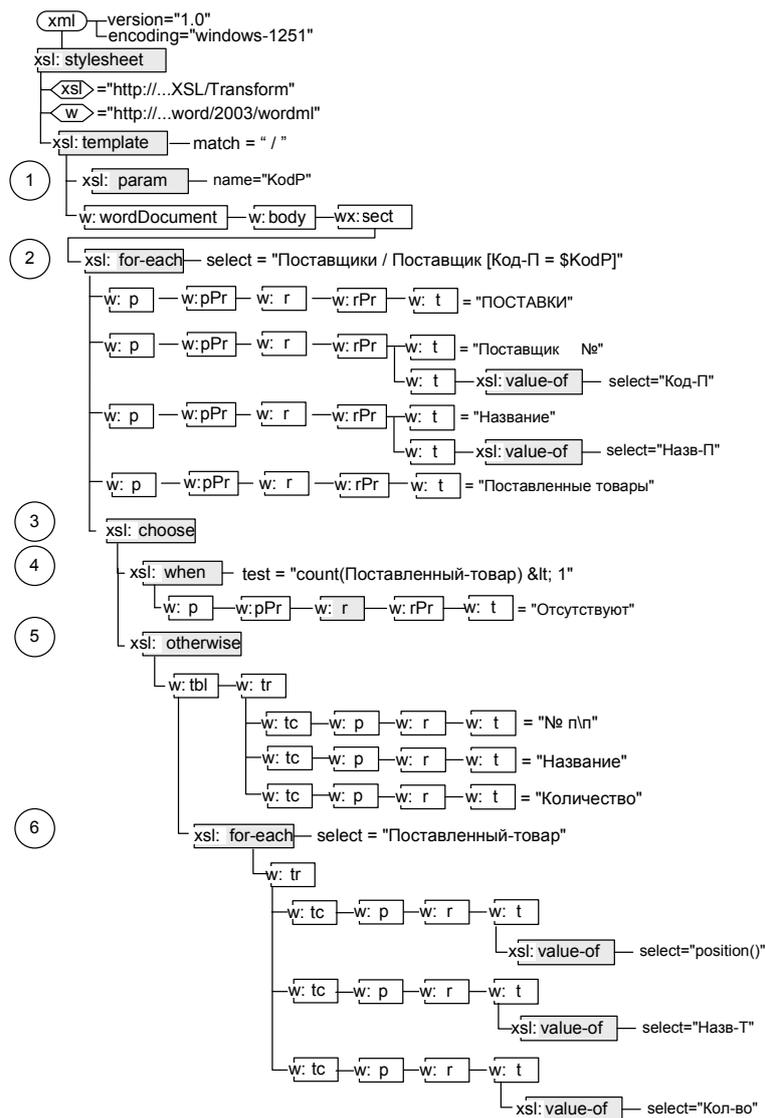


Рис. 8. Модель таблицы стилей, реализующей подстановочную модель

«Поставщик». Последний XML-элемент, в свою очередь, содержит вложенный XML-элемент «Поставленный-товар» с элементами «Код-Т», «Назв-Т» и «Кол-во», соответствующими одноименным реквизитам модели БР.

Порождение экземпляров ПЭД на основе модели класса хорошо «укладывается» в общую процедуру XSL-трансформации [2], если этот процесс рассматривать как процесс трансформации XML-базы реквизитов в результирующий XML-документ. В этом случае модель класса ПЭД необходимо реализовать в виде таблицы стилей (stylesheet) XSL-трансформации.

В этом подходе может быть учтен внешний запрос на персонализацию. В XSLT предусмотрена возможность параметрических таблиц стилей, зависящих от внешних параметров. При передаче параметра таблице стилей преобразование выполняется только для тех

данных из базы реквизитов, которые соответствуют запросу.

Так, к примеру, на рис. 8 показана модель таблицы стилей, реализующей преобразование реквизитов в документ, макет которого приведен на рис. 6, а. В качестве внешнего параметра таблице стилей передается значение кода поставщика, соответствующее одноименному атрибуту модели БР (1 на рис. 8). В результате преобразование выполняется только для данных о тех поставщиках, значение кода которых соответствует этому параметру (2). Собственно формирование документа выполняется последовательным копированием элементов разметки WordML [3] в результирующее XML-дерево в качестве конечных литеральных элементов (КЛЭ). Построение списка товаров определяется числом элементов «Поставленный-товар», вложенных в соответствующий экзем-

пляр элемента «Поставщик» (3). Если таких элементов нет, то в результирующем документе формируется параграф «отсутствуют». В противном случае формируется таблица, предназначенная для отображения информации о товарах (5). Далее для каждого элемента «Поставленный-товар» строится строка таблицы, в которую подставляются соответствующие значения из БР (6).

Таким образом, главный вопрос состоит в том, как построить XSL-таблицу стилей, чтобы она соответствовала рассмотренной модели класса ПЭД. Обсуждение возможных путей решения этой задачи выходит за рамки данной статьи.

ЗАКЛЮЧЕНИЕ

В статье обсуждаются вопросы формирования персонализированных электронных документов в формате WordML на основе информации из базы XML-реквизитов. При этом представлены следующие результаты.

1. Концепция персонализированных электронных документов, основанная на теории построения электронных документов и систем персонализации. Концепция отличается тем, что с целью обеспечения гибкости и удобства ведения электронных документов их содержимое адаптируется к конкретным пользователям за счет подстановки пользовательских данных (реквизитов) в статические фрагменты макета документа.

2. Модель класса ПЭД, основанная на представлении документа в виде иерархии вложенных фрагментов. Модель отличается тем, что с целью обеспечения персонализации экземпляров документа на основе базы XML-реквизитов узлы иерархии включают 1) форматированные фрагменты, внутри которых предусмотрены точки подстановки реквизитов, ссылающихся на базу реквизитов, и 2) элементы управления, обеспечивающие установку контекста обработки фрагментов, выбор одного из нескольких фрагментов, циклическое копирование в соответствии с состоянием базы реквизитов.

3. Технология формирования персонализированных электронных документов, основанная на использовании XML и сопутствующих стандартов. Технология отличается тем, что с целью обеспечения генерации документов в формате Microsoft Word база реквизитов представляется в XML-формате, а макет конечного документа – в форме таблицы стилей XSL-трансформации, в результате чего про-

цедура формирования персонализированного документа сводится к XSL-трансформации пользовательских XML-реквизитов. Работоспособность технологии подтверждена на примере генерации персонализированных документов в формате WordML.

СПИСОК ЛИТЕРАТУРЫ

1. **Миронов, В. В.** Информационная технология персонализации электронных документов Microsoft Office в Web-среде на основе XML / В. В. Миронов, Г. Р. Шакирова, В. Э. Яфаев // Вестник УГАТУ. Сер. Управление, вычислительная техника и информатика. 2008. Т. 10, № 2 (27). С. 112–122.
2. **Миронов, В. В.** XML-технологии в базах данных / В. В. Миронов, Н. И. Юсупова. Уфа : УГАТУ, 2004. 182 с.
3. **Lenz, E.** Office 2003 XML / E. Lenz, S. Laurent, M. McRay. O'Reilly, 2004. 450 p.
4. **Binstock, A.** Beyond Post: Adobe Forms vs. InfoPath? / A. Binstock [Электронный ресурс] (assets.devx.com/adobe/14199.pdf).

ОБ АВТОРАХ



Миронов Валерий Викторович, проф. каф. автоматиз. систем упр-я. Дипл. радиоп физик (Воронежск. гос. ун-т, 1975). Д-р техн. наук по упр-ю в техн. сист. (УГАТУ, 1995). Иссл. в обл. моделей крит. ситуаций и ситуац. управления.



Шакирова Гульнара Равиловна, асс. той же каф. Дипл. инженер по АСОИУ (УГАТУ, 2005). Канд. техн. наук по динамическим XML-документам (УГАТУ, 2008). Иссл. в обл. XML-технологий.



Яфаев Виль Эмарович, асп. той же каф. Дипл. инженер по АСОИУ (УГАТУ, 2006). Готовит дис. по проблеме генерации электронных документов на основе XML-технологии.