

УДК 004.91

В. В. МИРОНОВ, Г. Р. ШАКИРОВА, В. Э. ЯФАЕВ**ИНФОРМАЦИОННАЯ ТЕХНОЛОГИЯ ПЕРСОНАЛИЗАЦИИ
ЭЛЕКТРОННЫХ ДОКУМЕНТОВ MICROSOFT OFFICE
В WEB-СРЕДЕ НА ОСНОВЕ XML**

Обсуждаются вопросы построения электронных документов Microsoft Office на основе персонализированных пользовательских данных. Приводятся алгоритмы получения пользовательских данных с помощью интерфейсных документов и последующей генерации персонализированных документов в Web-среде. *Электронный документ ; персонализация ; Web-среда ; XML-технологии*

Современный уровень развития информационных технологий характеризуется тенденцией доминирования электронных документов (ЭД) над традиционными бумажными носителями информации. Управление созданием и ведением таких документов осуществляется с помощью компьютерных систем, которые в условиях повсеместного распространения Web-технологий становятся Web-ориентированными.

Особенностью многопользовательской работы с ЭД определенного класса через Web-среду является наличие индивидуальных данных, которых определяют содержимое пользовательского экземпляра конечного документа. В этих условиях важно обеспечить эффективный механизм персонализации ЭД. Адаптации контента (содержимого) ЭД к каждому конкретному пользователю ставит целью максимально упростить работу пользователей с документами за счет того, что индивидуальная информация каждого пользователя будет зафиксирована в документе и избавит пользователя от необходимости заполнения шаблонов соответствующих документов «вручную».

Идея формирования ЭД на основе пользовательской информации отражена в особом классе электронных документов — персонализированных электронных документах (ПЭД). ПЭД представляет собой электронный документ, содержащий постоянную (общую) и переменную (индивидуальную) информацию, формируемую в соответствии с переданными пользователем данными. Персонализация таких документов должна осуществляться комбинированием информации, которая «известна» серверу (общие данные),

и информации, которая серверу «не известна» (индивидуальные данные). К документам этого класса относятся любые ЭД, содержащие макет, который должен быть заполнен пользовательскими реквизитами (заявки, анкеты, приглашения, сертификаты и т. д.).

Отметим, что на сегодняшний день задача разработки электронных Web-документов, «заточенных» под конкретного пользователя, частично решается с помощью систем управления Web-контентом (Content Engineering Systems) [1, 2], позволяющих определить информационное наполнение Web-узла для различных пользователей. Однако это техническое решение не распространяется на системы, выполняющие генерацию динамического контента в форме ПЭД, определяемого заданными пользователем индивидуальными параметрами.

Целью настоящей статьи является выработка концепции создания и использования ПЭД в Web-среде на основе современных информационных технологий, таких как XML, Web-сервисы и т. д.

1. ПЕРСОНАЛИЗИРОВАННЫЕ ЭЛЕКТРОННЫЕ ДОКУМЕНТЫ И ТРЕБОВАНИЯ К НИМ

В зависимости от возможностей, которые предоставляются пользователю для работы с результирующим документом, можно выделить две группы ПЭД:

- готовые документы, представляющие собой конечный информационный продукт, не доступный пользователю для модификации;
- документы-заготовки, содержимое которых может быть дополнено пользователем

в контексте решаемой задачи. К ним относятся такие персонализированные документы, которые не могут быть использованы пользователем в качестве конечного документа, а их самостоятельное применение предполагает внесение дополнительной информации. В отличие от готовых ПЭД ПЭД-заготовки должны быть представлены в формате, допускающем дальнейшую модификацию содержимого ЭД. Поэтому применение таких «открытых» для редактирования форматов, как Microsoft Word и Excel предпочтительнее, чем использование «закрытых» форматов, таких как PDF.

Пользовательские данные, хранящиеся на сервере, могут быть дополнены в соответствии с информацией, занесенной пользователем в ПЭД-заготовку. Поэтому целесообразно выделить два вида ПЭД-заготовок:

- однократные, которые представляют собой окончательные ЭД, формируемые на основании данных, хранящихся на сервере, и передаваемые пользователю для последующей работы. Заполнение таких документов пользователями не предполагает внесение новой индивидуальной информации, которую нужно сохранить на сервере. При работе с однократным ПЭД процедура взаимодействия сервера с пользователем является однонаправленной;

- многократные, к которым относятся промежуточные документы, модифицируемые пользователем и передаваемые обратно на сервер. Они предназначены для получения дополнительной информации от пользователя и ее последующего сохранения на сервере. Следовательно, передача такого документа от сервера к пользователю является двунаправленной: пользователь получает ПЭД с сервера, вносит в него данные и возвращает обратно на сервер. Например, при создании анкет формируется ПЭД, передаваемый пользователю. В предусмотренные в документе поля пользователь вносит свои данные и передает документ обратно на сервер. Полученные данные сохраняются на сервере вместе с имеющейся информацией данного пользователя и могут быть использованы при генерации других ПЭД. Обработка многократных ПЭД-заготовок предполагает извлечение из документа дополнительных данных и обновление соответствующей пользовательской информации, хранящейся на сервере. Поэтому в таких документах должны быть предусмотрены элементы управления для ввода данных, которые могут быть однозначно выделены из общей структуры ЭД.

Процедура построения и применения однократных и многократных ПЭД-заготовок в Web-среде должна быть ориентирована на работу пользователя в режимах on-line и off-line взаимодействия с сервером. Это позволит освободить пользователя от необходимости единовременного ввода большого объема данных и модификации ПЭД в течение одного сеанса соединения с сервером.

В обобщенном виде процедура работы с ПЭД-заготовками представляется как последовательность следующих этапов (рис. 1):

1. Разработка макета многократной ПЭД-заготовки, в которой наряду с заданной пользовательской информацией и общими данными должны быть предусмотрены элементы управления для ввода индивидуальных пользовательских данных;

2. Формирование персонализированного пользовательского экземпляра многократной ПЭД-заготовки на основании данных, хранящихся на сервере, в редактируемом формате (например, Microsoft Word) и передача его пользователю;

3. Заполнение пользователем экземпляра многократной ПЭД-заготовки;

4. Разработка макетов однократных ПЭД-заготовок, в которых наряду с общей (постоянной) информацией предусмотрено наличие индивидуальной (переменной) информации;

5. Формирование пользовательских экземпляров однократных ПЭД с индивидуальной пользовательской информацией, хранящейся на сервере, в редактируемом формате (например, Microsoft Word) и передача его пользователю;

6. Модификация пользователем полученных экземпляров ПЭД-заготовок с целью доведения до уровня конечных документов.

Каждый из этих этапов может быть реализован с помощью современных Web-ориентированных программных механизмов.

2. ПОЛУЧЕНИЕ ДАННЫХ НА ОСНОВЕ ИНТЕРФЕЙСНЫХ ДОКУМЕНТОВ

Основным условием корректной персонализации ЭД является наличие на сервере персональной информации соответствующего пользователя. Процедура получения такой информации является самостоятельной задачей, решение которой определяется характером и объемом запрашиваемых данных.

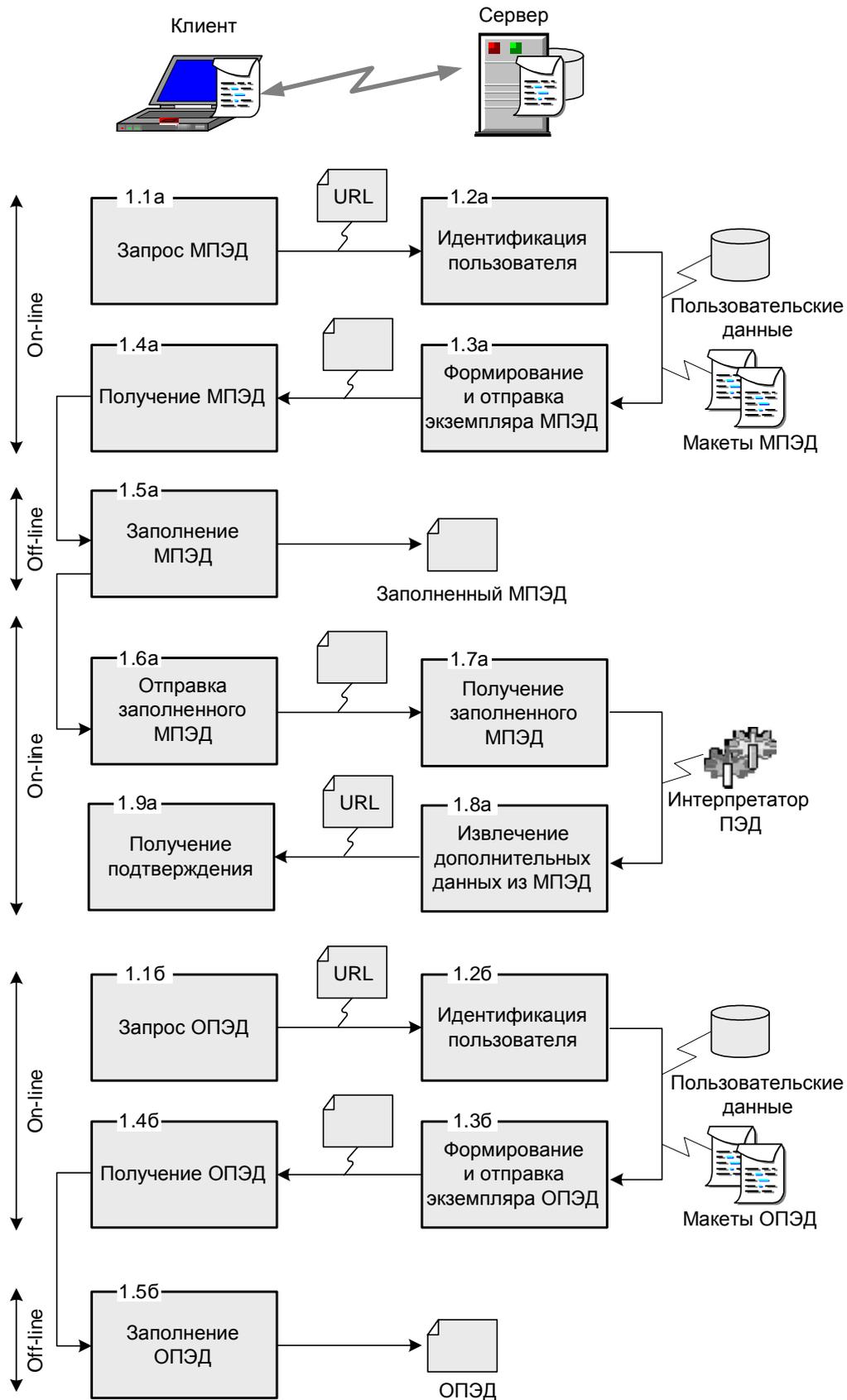


Рис. 1. Обобщенная схема процедуры создания и применения многократных (а) и однократных (б) персонализированных электронных документов: ПЭД – персонализированный электронный документ, МПЭД – многократный ПЭД; ОПЭД – однократный ПЭД

Традиционный подход к получению пользовательских данных основан на использовании механизма Web-форм, содержащих элементы управления для ввода индивидуальных пользовательских параметров. Пользователь в режиме on-line работы с сервером последовательно вносит данные в соответствующие поля ввода формы, а затем передает ее на Web-сервер.

Механизм диалогового воздействия с пользователем на основе форм широко распространен в современных Web-ориентированных системах, что обусловлено, прежде всего, простотой его реализации и поддержкой ведущими производителями программного обеспечения. Вместе с тем, применение Web-форм оправданно только для решения задач, предполагающих работу с простыми в количественном и качественном отношении пользовательскими данными. Усложнение задачи получения данных приводит к значительному увеличению числа расположенных на форме полей для заполнения, что существенно усложняет работу пользователей Web-узла.

Еще один недостаток механизма Web-форм связан с жесткой привязкой к режиму on-line взаимодействия пользователя с сервером. Однако все запрашиваемые данные не всегда могут быть введены пользователем одновременно, а отложенное заполнение формы может привести к частичной потере уже введенной информации. В этих условиях важно обеспечить пользователю возможность вводить данные в режиме off-line, а затем передавать их на сервер в режиме on-line.

Современные информационные технологии позволяют частично решить проблему автономного ввода данных для их последующей передачи на сервер. Например, такие механизмы, как Adobe Forms и Microsoft InfoPath [3], реализуют клиентский интерфейс в форме «интеллектуального документа». Он представляет собой основанный на технологии Web-форм XML-ориентированный ЭД, позволяющий пользователю вводить любые данные в автономном режиме, независимо от наличия соединения с сервером. Внешнее представление «интеллектуального документа» может быть настроено так, чтобы максимально повторить соответствующий бумажный аналог, что позволит существенно упростить работу пользователя. Внутреннее представление пользовательских данных в «интеллектуальном документе» задается в XML-формате, что обеспечивает их гибкость, открытость и платформонезависимость.

Вместе с тем технология «интеллектуальных документов» в применении к механизму получения данных от пользователя характеризуется рядом недостатков, основным из которых является необходимость установки соответствующего клиентского программного обеспечения, что связано с приобретением дорогостоящих лицензий. В ходе исследования этих механизмов наблюдалась недостаточная интеграция с продуктами сторонних фирм. Помимо этого для работы с таким программным обеспечением пользователь должен обладать соответствующими навыками, что неприемлемо в условиях единовременного применения «интеллектуального документа». В связи с этим возникает актуальная задача разработки нового технического решения, сочетающего в себе преимущества «интеллектуальных документов» и интерфейс, хорошо знакомый пользователям.

Форматы Microsoft Office, в первую очередь Word и Excel, являются наиболее популярными на сегодняшний день способами представления ЭД, а текстовый процессор Microsoft Word установлен практически на каждом персональном компьютере. Среди достоинств форматов Microsoft Word и Excel помимо широкого распространения можно отметить и наличие механизмов, позволяющих получать и обрабатывать отдельные данные, введенные пользователем в документ, например, пользовательские формы. При этом применение сложного форматирования позволяет добиться полного сходства ЭД с соответствующей формой на бумажном носителе.

Обсудим в этом плане возможности формата Word. Документ Microsoft Word, снабженный пользовательской формой с элементами управления, можно использовать как клиентский интерфейс для получения данных от пользователя. С одной стороны, он содержит элементы Web-формы для ввода данных, а с другой — поддерживает возможность работы пользователя в автономном режиме. Эта концепция отражена в особом классе документов формата Microsoft Word, названных интерфейсными документами (ИД). ИД представляет собой электронный документ в формате MS Word, содержащий поля ввода для пользовательских данных (переменной информации) и элементы форматирования (постоянную информацию), определяющую общую структуру документа.

Обозначим переменную информацию ИД как базу пользовательских реквизитов (БПР). Тогда задача получения пользовательских данных сводится к извлечению БПР

из ИД. Для этого с сервера на клиентский компьютер передается экземпляр соответствующего ИД. Пользователь заполняет полученный документ индивидуальными данными и передает их обратно на сервер, где интерпретатор ИД извлекает из него БПР. В этой связи важно обеспечить однозначную идентификацию БПР ИД, а именно — каждого поля ввода в документе.

Текстовый процессор Microsoft Word поддерживает два основных механизма идентификации полей ввода: закладки (Bookmarks) и разметка документа XML-тегами на основе прикрепленной схемы. Первый вариант предполагает сопоставление с каждым пользовательским полем ввода отдельной именованной закладки, последующее обращение к которой выполняется на основе объектной модели Microsoft Word. БПР, извлеченная из такого документа, представляет собой множество переменных, задаваемых программно. Недостатком такого подхода является большая трудоемкость реализации документа и его интерпретатора, связанная с необходимостью программирования каждого элемента и жесткой привязкой интерпретатора к заданному множеству закладок. Кроме того, необходимость обработки документов Word на сервере неоптимальна как в плане безопасности, так и производительности.

Разметка документа XML-тегами позволяет поставить в соответствие каждому полю ввода собственный XML-элемент. Внутренняя структура такого документа представлена разметкой двух видов: системной, содержащей инструкции форматирования в формате WordML, и пользовательской, описывающей введенные данные [5]. Такой подход является наиболее эффективным, поскольку XML-представление пользовательских данных позволяет обеспечить их платформонезависимость, объектность и открытость, что немаловажно для их эффективной программной обработки.

Аналогичным образом идентификация пользовательских данных может быть выполнена и в электронной таблице Microsoft Excel. В результате внутреннее представление таких файлов задается системной разметкой, задающей структуру каждого листа книги, включая стили форматирования, и пользовательской разметкой, описывающей введенные данные [6].

Применение XML-ориентированных форматов Microsoft Word и Excel сводит задачу извлечения БПР к отделению пользовательской разметки от системной, которая состо-

ит в необходимости “фильтрации”, “очистки” пользовательских данных от системной информации. Открытость XML-формата позволяет выполнять обработку подобных документов с помощью стандартных программных средств, обеспечивающих механизмы для работы с XML-данными. Структурная разметка, заданная в определениях схем WordML и SpreadsheetML, не отделена от пользовательской информации и уникальна для каждого экземпляра документа. Как показал опыт, универсальное обращение к персональным данным напрямую, с помощью запросов типа XPath невозможно. В результате процедура извлечения пользовательских данных будет жестко привязана к экземпляру обрабатываемого документа, что неэффективно как в плане универсальности обработки данных, так и производительности Web-узла в целом.

Следовательно, возникает **проблема «очистки» пользовательских данных от системной информации** документа Microsoft Word или электронной таблицы Excel. Рассмотрим решение этой задачи для документов Microsoft Word.

Текстовый процессор Microsoft Word располагает механизмом удаления форматирования Word, основанным на сохранении WordML-документа с опцией «Сохранить только данные». Однако документ, полученный в результате такого сохранения, будет содержать не только XML-элементы, определенные присоединенной к документу Word схемой, но и текстовую информацию, расположенную за пределами этих элементов. Следовательно, сформированная XML-БПР помимо пользовательских данных будет содержать «лишнюю» информацию из соответствующего документа Word. Хранить такую информацию на сервере нецелесообразно, поэтому необходимо новое решение, позволяющее извлечь из документа Word только пользовательские данные, определенные присоединенной к документу схемой.

Для решения этой задачи предлагается применить к документу XSL-преобразование на основе инструкций форсирующей трансформации, которые анализируют XML-структуру данных и удаляют все элементы, попадающие в разряд системных. Для реализации такого преобразования была разработана таблица стилей, которая содержит описание системных элементов WordML-разметки, которые должны быть удалены из результирующего множества данных. Для этого были использованы девять префиксов пространств имен, предусмотренных схемой WordML.

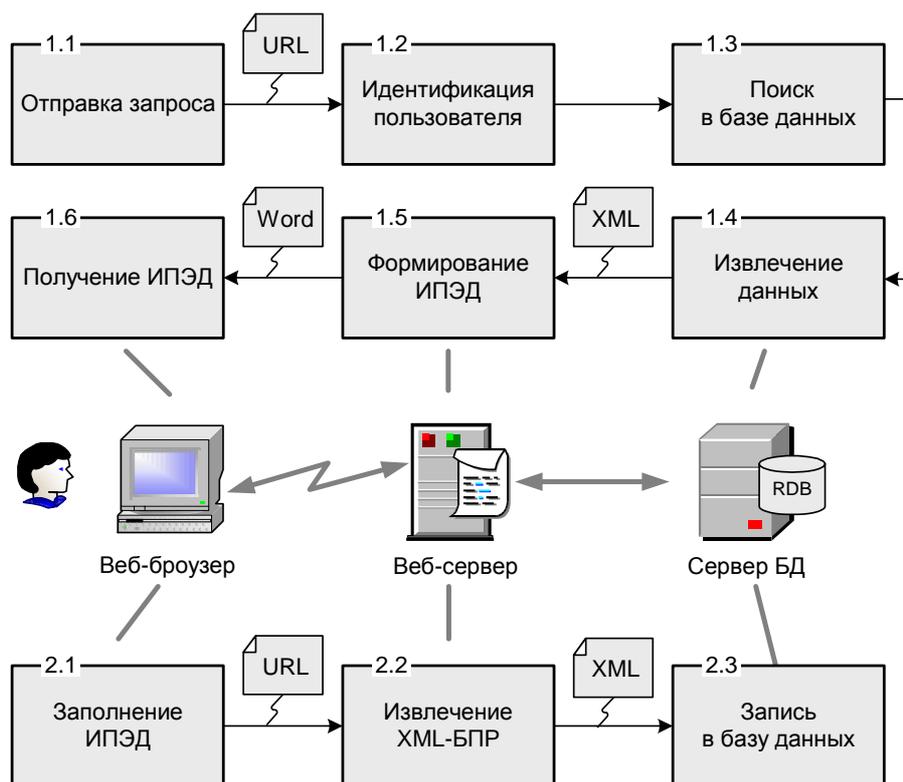


Рис. 2. Получение пользовательских данных на основе ИПЭД

Все XML-элементы с таким префиксом являются системными элементами WordML-разметки, поэтому для указания в инструкциях трансформации всего множества служебных данных был использован формат «префикс пространства имен:*», например, w:* . Для каждого из этих элементов применяется инструкция, выполняющая их удаление.

Для выполнения преобразования XSL-таблица стилей размещается на сервере, а каждый полученный от пользователя ИД обрабатывается в первую очередь в соответствии с указанными в ней инструкциями. Результатом выполнения такого XSL-преобразования станет набор XML-элементов пользовательской разметки, или БПР в формате XML (XML-БПР).

Один и тот же ИД может быть использован пользователем для многократного изменения собственных данных, хранящихся на сервере. Для этого ИД должен быть реализован в классе МПЭД. Документы такого класса называются интерфейсными МПЭД (ИМПЭД). Применение ИПЭД особенно удобно тогда, когда на сервере уже имеются данные соответствующего пользователя. Для их хранения может быть использована реляционная база данных (РБД), расположенная на сервере. Для размещения пользовательской информации в РБД может быть выделено ли-

бо несколько полей соответствующих таблиц, либо единственное поле типа XML для хранения XML-данных.

Обобщенная схема процедуры получения пользовательских данных на основе ИПЭД приведена на рис. 2.

По запросу к РБД формируется XML-документ, содержащий пользовательские данные. К этому документу применяется XSL-преобразование, на основе которого формируется экземпляр ИПЭД, передаваемый пользователю. При этом пользователь может как занести в ИПЭД новые данные, так и внести изменения в уже имеющуюся информацию. После передачи ИПЭД на сервер происходит обновление пользовательских данных в РБД.

3. ФОРМИРОВАНИЕ ПЕРСОНАЛИЗИРОВАННЫХ ДОКУМЕНТОВ

Разработка персонализированных документов базируется на заполнении электронных макетов индивидуальными пользовательскими данными. В качестве макетов должны выступать программные механизмы, позволяющие выполнять автоматическую подстановку параметров из БПР. Кроме того, макеты должны допускать дальнейшую обработку пользователем с целью доведения заготовки до уровня конечного доку-

мента. Следовательно, для реализации макета следует использовать объектный, «открытый» для редактирования формат, такой как Microsoft Word.

Текстовый процессор Microsoft Word поддерживает несколько механизмов вставки данных в документы: закладки, свойства документа, коды полей и т.д. Вместе с тем такой подход требует организации процедуры обработки документа на сервере, что снижает эффективность и безопасность работы Web-узла. Другой механизм объектной вставки данных в Word предполагает применение иерархии XML-узлов. В этом случае документ представляет собой множество XML-элементов, благодаря чему приложение просто анализирует XML-данные, чтобы обнаружить соответствующие объекты документа и работать с ними. В этой связи целесообразно реализовывать персонализированные документы класса ОПЭД в форме XML-ориентированных ИД.

В обобщенном виде процедуру персонализации ИД можно представить как преобразование множества пользовательских данных в персонализированный документ-заготовку. Каждый документ Microsoft Word может быть представлен как совокупность системных XML-тегов форматирования и XML-элементов с пользовательской информацией. Следовательно, возникает задача формирования персонализированного XML-документа, включающего системные и пользовательские элементы для его последующего преобразования в формат doc.

Обобщенная схема выполнения на сервере процедуры формирования персонализированного документа приведена на рис. 3.

При запросе пользователем документа на сервере формируется поток XML-данных, содержащий индивидуальные пользовательские данные (XML-БПР). В зависимости от организации соответствующих данных XML-поток формируется на основе либо схем сопоставления (Mapping Schema)¹, либо стандартных запросов на выборку (SELECT ... FROM)². И в том, и в другом случае результатом обращения к РБД является поток данных, структурированных с помощью XML-разметки.

XML-поток, сгенерированный на сервере, передается на клиентский компьютер как файл, соответствующий определениям XML Schema, заданными в Word. Для этого вы-

полняется XSL-трансформация, реализуемая в форме таблицы стилей (XSL Style Sheet). В ней предусмотрены программные инструкции, задающие множество служебных элементов общей структуры документа и правила его форматирования в контексте Word XML Schema. Для внесения индивидуальной информации в таблицу стилей предусмотрены инструкции извлекающей трансформации, содержащие XPath-запросы к потоку XML-БПР.

Применение технологии XSL-трансформации XML-данных в формат документа Microsoft Word позволяет организовывать работу Web-узла в условиях отсутствия на сервере исполняемых файлов Microsoft Word, так как благодаря поддержке XML все операции генерации документа выполняются на уровне XML-разметки. В результате работа пользователя с ИПЭД осуществляется на локальном компьютере, полностью исключая необходимость обработки документа Word на сервере.

Программная логика Web-узла оперирует процедурой формирования и трансформации XML-потока на уровне хранимых в памяти DOM-объектов. DOM-объект (W3C Document Object Model). DOM представляет XML-данные как дерево, хранящееся в памяти, с заданным корневым элементом. Это позволяет оперировать иерархией XML-данных как программным объектом, применяя к нему операции создания, редактирования и трансформации.

В процессе генерации запрашиваемого пользователем ИПЭД последовательно формируются три DOM-объекта: для исходного потока XML-БПР, для таблицы стилей (которая также является корректным XML-документом) и для результирующего документа в формате WordML. Все последующие операции выполняются только в рамках созданных DOM-объектов. Следовательно, на каждом этапе процедуры генерации ИПЭД Web-приложение оперирует виртуальным XML-документом, последовательно преобразуя его к конечному формату документа Microsoft Word. Это позволяет существенно снизить аппаратные затраты пользователя, связанные с физическим хранением множества временных XML-файлов: все операции выполняются на уровне виртуального пользовательского потока данных. В результате сама процедура генерации ИПЭД представляет собой циркуляцию на сервере персонализированных потоков данных.

¹Схема сопоставления представляет собой XML-документ класса XML Schema, структура которого описывает иерархию XML-элементов и XML-атрибутов, сопоставленных с таблицей или полем реляционной базы данных [4].

²Применение запроса вида SELECT ... FROM к поле типа XML возвращает результат в формате XML.

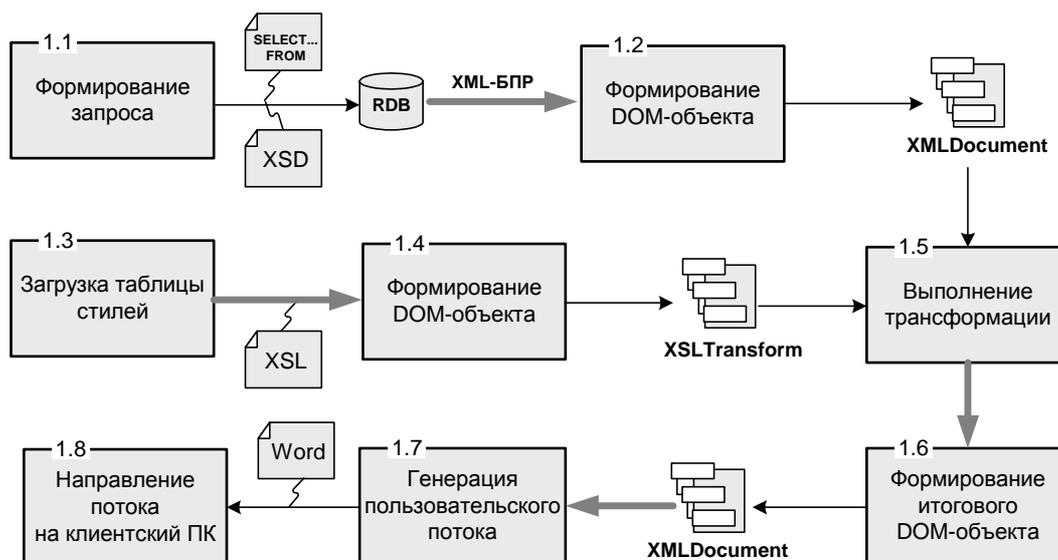


Рис. 3. Обобщенная схема генерации персонализированного документа на сервере

На начальном этапе в DOM-объект заносится XML-поток XML-БПР, формируемый на основании запроса к РДБ, расположенной на сервере. В другой DOM-объект вносится иерархия XML-элементов, составляющих таблицу стилей, файл с которой хранится на сервере.

На уровне программной реализации (например, на основе технологии ASP.NET) формируемые DOM-объекты относятся к разным классам. Так, объект с XML-БПР реализуется в классе `XmlDocument`, а объект с таблицей стилей — в классе `XSLTransform`. Оба класса предназначены для работы с DOM-объектами, но предоставляют разные методы и свойства в соответствии со спецификой представляемых XML-данных. Так, к примеру, в классе `XSLTransform` доступен метод `Transform`, который предназначен для выполнения XSL-преобразования заданного XML-документа или DOM-объекта. Применим этот метод для трансформации XML-БПР в контексте расположенной на сервере таблицы стилей.

Результат выполнения трансформации XML-БПР в соответствии с заданной таблицей стилей заносится в DOM-объект, реализуемый в классе `XmlDocument`. Он содержит окончательную XML-разметку результирующего документа, включая инструкции форматирования и индивидуальные пользовательские данные.

Последний этап преобразования — передача пользователю файла с содержимым итогового DOM-объекта. Для этого на клиентский компьютер передается поток данных, к которому в режиме `attachment` прикрепляется файл с расширением `doc`. Контент этого файла

формируется из содержимого итогового DOM-объекта. В результате пользователь получает файл, состоящий из XML-данных, однако, при этом его можно загружать и напрямую использовать из приложения Microsoft Office. При этом документ уже снабжен индивидуальной информацией, и пользователю остается только в режиме `off-line` довести его до уровня конечного документа.

4. ФОРМИРОВАНИЕ ШАБЛОНА ПРЕОБРАЗОВАНИЯ XML-WORD

Для генерации персонализированного документа пользовательские данные, представленные в XML-формате, должны быть преобразованы в документ Microsoft Word. Для этого необходим шаблон преобразования, содержащий правила заполнения постоянной составляющей документа переменной пользовательской информацией.

Наиболее эффективным механизмом преобразования XML-данных в новый формат является XSL-трансформация. Применение XSL-трансформации выполняется с помощью XML-файлов специального вида — таблиц стилей, содержащих набор правил преобразования исходного XML-документа, написанных на языке XSLT [4].

Таблица стилей может быть разработана «вручную» — последовательным применением XML-элементов, предусмотренных схемой `WordML`. При этом для указания расположения переменной пользовательской информации используются инструкции форсирующей и извлекающей трансформации, содержащие XPath-описание XML-данных, к кото-

рым должно быть применено преобразование. В результате шаблон документа Word строится путем комбинирования WordML-элементов и пользовательских инструкций трансформации. Вместе с тем такой подход применим только для простых документов Word, не содержащих сложных элементов форматирования и структур данных. Причиной такой ограниченности «ручного» программирования таблицы стилей является необходимость предварительного изучения определений WordML-схемы, содержащей описание значительного числа XML-элементов, большая часть которых содержит элементы сложного форматирования документа (например, элементы построения таблиц).

Другой подход к построению шаблона преобразования XML-Word основан на использовании XML-возможностей текстового процессора Microsoft Word. Проектирование таблицы стилей начинается с разработки прототипа шаблона документа, в котором задается порядок расположения постоянной и переменной информации и элементы их форматирования. При этом для указания переменной информации используется система меток, позволяющих идентифицировать каждый пользовательский параметр. В качестве меток могут выступать слова, легко выделяемые в контексте всего множества данных. При этом важно обеспечить уникальность каждой метки как среди всех меток, так и на уровне элементов постоянной информации.

На следующем этапе разработанный прототип необходимо преобразовать в формат XML. Для этого при сохранении документа следует выбрать опцию «Тип файла → XML-документ». В результате будет получен XML-документ, содержащий необходимые элементы WordML-разметки — XML-прототип шаблона.

При преобразовании XML-прототипа в таблицу стилей нужно выполнить корректировку инструкций по обработке документа. Для этого строку «<mso-application progid="Word.Document"?>» разметки XML-прототипа следует заменить на инструкции XSL-обработки «<xsl:stylesheet version="1.0" xmlns:xsl="http://www.w3.org/1999/XSL/Transform">». В результате XML-прототип будет рассматриваться XML-анализатором как XSL-таблица стилей. Однако при удалении инструкции типа «mso-application» будет потеряна информация о типе целевого приложения, в нашем случае — Microsoft Word. В результате применение трансформации приведет к созданию XML-файла с элементами

WordML, а не документа Microsoft Word с элементами XML-разметки.

В соответствии с синтаксисом XSL-трансформации для указания целевого приложения используется элемент «xsl:processing-instruction». В этой связи XML-прототип документа должен быть дополнен элементом «<xsl:processing-instruction name="mso-application"> <xsl:text>progid="Word.Document"</xsl:text></xsl:processing-instruction>», который создаст в результирующем дереве узел инструкции по обработке «<mso-application progid="Word.Document"?>». Инструкция по обработке должна быть применена ко всему документу, следовательно, элемент «xsl:processing-instruction» должен быть применен в рамках инструкции форсирующей трансформации «<xsl:template match="/>».

Полученная в результате описанных преобразований таблица стилей представляет собой XML-прототип документа Word. Для доведения программного кода до уровня шаблона его следует дополнить инструкциями подстановки пользовательских данных в результирующий документ. Для этого используются команды извлекающей трансформации типа «<xsl:value-of select>», с помощью которых из пользовательских XML-данных (XML-БПР) извлекается значение элементов или атрибутов в соответствии с заданным XPath-выражением. В разметке XML-прототипа элементы «<xsl:value-of select>» заменяют метки переменной информации, заданные на начальном этапе разработки. В результате переменная информация полученного документа будет представлена как результат выполнения инструкций извлекающей трансформации из XML-БПР и будет уникальна для каждого экземпляра документа.

Полученный таким образом шаблон трансформации пользовательских XML-данных в документ Word представляет собой XSL-таблицу стилей, содержащую служебную WordML-разметку и инструкции подстановки пользовательских данных. Каждый шаблон задает класс документов, персонализированные экземпляры которого передаются пользователю.

Разработка шаблона трансформации с применением средств поддержки XML в Microsoft Office позволяет задавать шаблоны сколь угодно большой сложности и объема, поэтому этот вариант является более эффективным по сравнению с «ручным» программированием таблицы стилей.

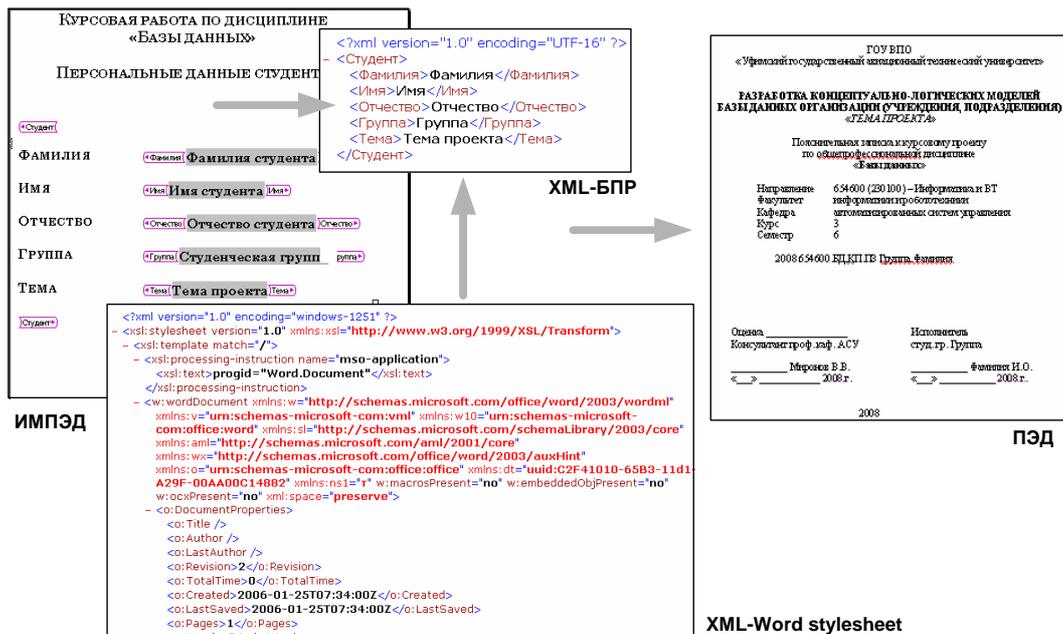


Рис. 4. Практическая иллюстрация применения интерфейсных документов для получения пользовательских данных и формирования персонализированных документов

5. ПРИМЕНЕНИЕ ПЕРСОНАЛИЗАЦИИ ЭЛЕКТРОННЫХ ДОКУМЕНТОВ

Для иллюстрации практического приложения концепции применения ИД для формирования персонализированных документов рассмотрим процедуру формирования заготовки пояснительной записки для студентов, выполняющих курсовую работу по дисциплине «Базы данных».

Для каждого студента индивидуальной информацией является фамилия, имя, отчество, группа и тема курсовой работы. Для передачи этих данных на сервер студент получает интерфейсный документ, содержащий поля ввода соответствующих параметров (рис. 4). Студент открывает документ с помощью текстового процессора Microsoft Word и заполняет его в режиме off-line, после чего передает его обратно на сервер. Полученный документ обрабатывается интерпретатором ИД, который извлекает из него индивидуальные пользовательские данные в формате XML – XML-БПР (рис. 4). Полученные данные заносятся в расположенную на сервере базу данных.

На следующем этапе студент выбирает тип документа, который должен быть сгенерирован, например, титульный лист пояснительной записки. Интерпретатор загружает расположенный на сервере файл трансформации XML-данных в формат документа Microsoft Word – таблицу стилей XML-Word Stylesheet (рис. 4). Из базы данных в формате XML

извлекается персонализированная пользовательская информация, к которой применяется XSL-преобразование на основе загруженной таблицы стилей. В результате формируется новый объект – документ Microsoft Word, содержащий заготовку титульного листа пояснительной записки. В документ загружена индивидуальная информация данного студента – фамилия, имя, отчество и т. д. Студенту остается только внести некоторые коррективы по своему усмотрению для доведения переданной ему заготовки до уровня конечного документа.

Аналогичным образом персонализируются электронные документы, соответствующие разделам пояснительной записки в контексте этапов курсового проекта. В дальнейшем на основе персональных данных каждого студента формируется персонализированный документ, в котором содержатся элементы задания на курсовой проект и личные данные студента. Каждый такой документ представляет собой заготовку, в которую студент вносит свои проектные решения и доводит таким образом весь документ до уровня конечной пояснительной записки.

ЗАКЛЮЧЕНИЕ

Тенденции развития современных информационных технологий характеризуются разработкой механизмов, решающих задачу оптимизации формирования электронных документов. Один из аспектов этой задачи свя-

зан с персонализацией электронного документа, т. е. адаптацией электронного документа к конкретному пользователю.

Персонализированный электронный документ формируется путем заполнения заданного макета индивидуальной пользовательской информацией. Предложено реализовывать запрос пользовательских данных с помощью интерфейсных документов, представляющих собой документы Microsoft Word и Excel, содержащие поля для ввода данных, идентифицируемые XML-тегами на основе прикрепленной схемы. В этой связи возникает задача извлечения пользовательских данных из интерфейсного документа. Для ее решения предложена таблица стилей XSL-преобразования, содержащая инструкции формирования результирующего XML-документа только из XML-элементов, предусмотренных прикрепленной к документу Word схемой. XML-документ, получаемый в результате такого преобразования, носит название базы пользовательских реквизитов.

Для формирования персонализированного документа на основе XML-базы пользовательских реквизитов предлагается использовать макеты, организованные в форме таблиц стилей, содержащих инструкции XSLT-преобразования XML-данных в формат документа Microsoft Word. Для решения задачи разработки шаблона преобразования XML-Word предложено решение, основанное на встроенных механизмах текстового процессора Microsoft Word, состоящее в разработке прототипа шаблона документа и его последующем переводе в XSL-формат. Такой подход позволяет перенести обработку документа Word с сервера на компьютер пользователя, что существенно повысит производительность и безопасность Web-приложения.

На основании представленных концептуальных положений предложено Web-решение на основе XML-технологий, динамически создающее персонализированные документы Microsoft Office, доведенное до уровня информационной технологии. Практическая реализация технологии для комплекса документации курсового проекта показала работоспособность и эффективность предложенного подхода.

СПИСОК ЛИТЕРАТУРЫ

1. **Internet.** Content Engineering [Электронный ресурс] (<http://www.pcpage.com/contenteng/contenteng.asp>).
2. **Internet.** Контент-инжиниринг [Электронный ресурс] (<http://www.wdigest.ru/content-engineering>).
3. **Binstock, A.** Beyond Post: Adobe Forms vs. InfoPath? / A. Binstock [Электронный ресурс] (assets.devx.com/adobe/14199.pdf).
4. **Миронов, В. В.** XML-технологии в базах данных / В. В. Миронов, Н. И. Юсупова. Уфа : УГАТУ, 2004. 182 с.
5. **Internet.** Overview of WordprocessingML [Электронный ресурс] (http://rep.oio.dk/Microsoft.com/officeschemas/wordprocessingml_article.htm).
6. **Internet.** Software Engineering: SpreadsheetML [Электронный ресурс] (<http://dsmyth.blogspot.com/2006/03/spreadsheetml.html>).

ОБ АВТОРАХ



Миронов Валерий Викторович, проф. каф. автоматиз. систем упр-я. Дипл. радиофизик (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 Равилевна, аспирант той же каф., асс. каф. вычислит. математики и кибернетики. Дипл. инженер по АСОиУ (УГАТУ, 2005). Готовит дис. в обл. использования встроенных моделей в XML-документах.



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