УДК 519.24

А.Н. КУРБАЦКИЙ, В.А. ЧЕУШЕВ, БИНЬ СЮЕ (КНР), С.В. ГУРНОВСКАЯ

АВТОМАТИЧЕСКАЯ ГЕНЕРАЦИЯ И МАКЕТИРОВАНИЕ ДОКУМЕНТОВ НА ОСНОВЕ XML-ТЕХНОЛОГИИ

A complex of methodical and technological solutions is proposed for automatic generation and prototyping in various forms of on-line documents, representing dynamic virtual structures of an information resource, which information system builds to meet user's needs. Prototyping procedures are developed and approved to obtain documents in HTML and for Microsoft Word, Corel Ventura, LaTeX. Described technological process can be used also to generate dynamically data sources for representation in other applications and ActiveX components. As a whole, proposed solutions are intended for ensuring user's requirements on high-quality and editable documents, representing auto-generated information resource.

Способность представлять вторичные информационные ресурсы (ИР) в качественной форме – одна из главных характеристик систем управления информационными ресурсами (СУИР), поисковых, аналитических и других информационных систем (ИС) наряду с полным и точным удовлетворением информационной потребности пользователя (ИПП). Важно не только найти и обработать информацию, релевантную ИПП, но и представить результат в виде электронного документа, пригодного для анализа, сохранения пользователем и бумажной публикации [6].

В работе излагается основанный на применении XML-технологий комплекс решений, предназначенных для применения в различных ИС для генерации документов, представляющих вторичные ИР, автоматического их макетирования в разных форматах и передачи различным приложениям.

Предлагаемые решения могут применяться в разных «проектных исполнениях». Мы рассматриваем вариант их применения в web-приложении с «горячим» подключением текстового процессора и издательской системы на стороне клиента. В таком исполнении они включаются нами в разрабатываемую в Научно-исследовательской лаборатории информационно-технологических систем БГУ (НИЛИТС) технологию проектирования баз знаний и применяются при создании ряда ИС.

Исходные посылки

1. Будем считать, что удовлетворение ИПП обеспечивают *виртуальные документы*, не существующие статически, а получающие свое наполнение адекватно содержанию и моменту запроса из различных составляющих ИР (БД, XML-документов). Для класса виртуальных документов создается *программа генерации документа* (ПГД), способная формировать документы, общность которых определяется назначением и содержанием ПГД, различия – ее входными параметрами.

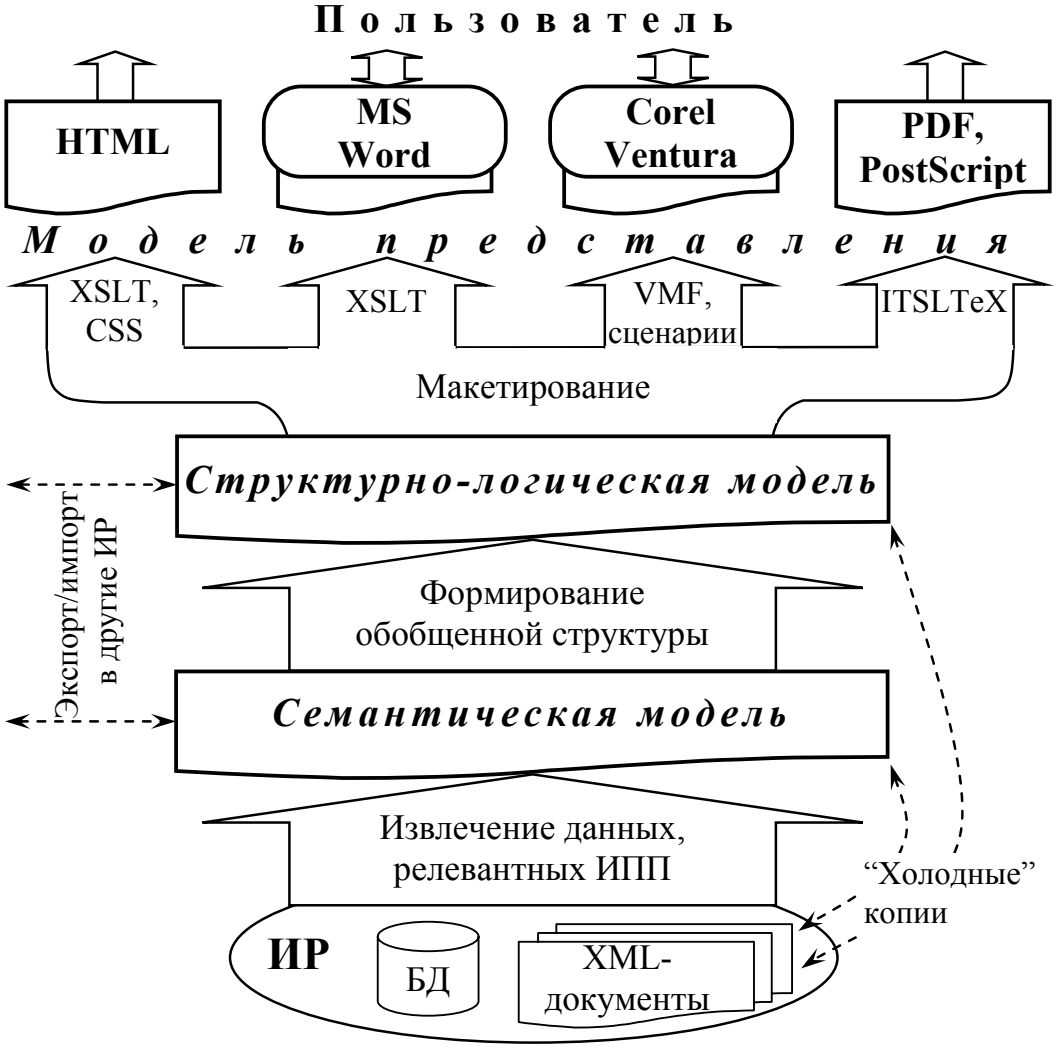
2. Для многих приложений недостаточно генерировать HTML-версии для печати, как в большинстве Интернет-систем: желательно, а зачастую обязательно обеспечить возможность редактирования пользователем полученных документов, в ряде случаев требуется качественное их макетирование [3]. И то и другое не обеспечивается средствами HTML. Целесообразно макетировать документ средствами

приложений: текстовых процессоров, издательских систем. И если говорить о профессиональном уровне подготовки документа, то здесь не должно быть иллюзий, например, что качественный документ Corel Ventura можно автоматически получить путем конвертации из HTML или Word: практика не подтверждает этого.

3. Часто необходимо обеспечить сохранение «холодной» копии виртуального («горячего») документа, полученного конкретным запросом в конкретный момент. Копирование во всех предусмотренных форматах неэффективно, лучше сохранить в ИР некую базовую форму, которая может быть а) в нужный момент воспроизведена с приведением к требуемому формату, б) экспортирована в другие ИР.

Общая модель формирования документа

В общем виде спроектированная, апробированная и применяемая нами при создании ряда практических систем схема формирования и макетирования документа представлена на рисунке. С момента извлечения из ИР до передачи приложению данные существуют в XML-форме, проходя несколько преобразований. Характеризуя модель документа на каждой стадии, мы говорим о его XML-разметке.



Общая схема автоматической генерации и макетирования виртуального документа

1. *Семантическая модель* (*СМ*). На первой стадии ПГД (назовем эту подпрограмму ПГД-СМ) извлекает и обрабатывает адекватно запросу первичные компоненты ИР и формирует документ, XML-разметка которого именует элементы данных по смыслу и задает их структуру, не неся информации о том, где и как они должны представляться в итоговом документе. СМ компактна, ее целесообразно использовать для сохранения «холодных» копий и экспорта. Сохраненная копия становится первичным компонентом ИР, реальным экземпляром виртуального документа в семантической его составляющей. Понятно, что множество ПГД-СМ безгранично: только в конкретном случае можно говорить о том, какие компоненты какого ИР ПГД-СМ извлекает и как обрабатывает. Соответственно разметка СМ произвольна по словарю. Из СМ можно получить различные по структуре и оформлению документы.

2. *Структурно-логическая модель* (*СЛМ*), или *обобщенная модель представления,* формируется из СМ посредством заданного преобразования (которых может быть несколько) и определяет структуру и логику данных в терминах документа (раздел, подраздел, таблица и т. д.) без детализации стилей и конкретики представления в различных форматах. СЛМ также можно использовать для сохранения «холодных» копий и экспорта. Количество формально различных структурных единиц документов сравнительно невелико, поэтому разметка СЛМ уже не должна быть произвольной, ее словарь целесообразно в определенной области применения стандартизировать, чтобы многократно использовать созданные интерпретации разметки. Для этой цели в НИЛИТС разработан Язык структурно-логической разметки документов (ЯСЛРД), определяющий словарь разметки типовых элементов документов (структурные единицы, базовые элементы оформления, ссылки, таблицы и т. д.). Заметим, что перейти к СЛМ не обязательно означает потерять семантическую разметку – ее можно сохранить в атрибутах разметки. Из СЛМ можно получить разные по стилевому оформлению документы.

3. *Модель представления* (*МП*) в конкретном приложении (формате) формируется одной из процедур преобразования – *макетированием* (ПГД-МП), которое а) дополняет структурно-логическую разметку стилевой составляющей и б) реализует специфику представления документа в конкретном приложении (формате). Рассмотрим детали ее реализации.

Автоматическое макетирование документа

Нами созданы и апробированы процедуры макетирования документов для приложений и форматов HTML, Microsoft Word (версии, поддерживающие XML), Corel Ventura (версии, поддерживающие XML), LaTeX.

HTML. Для web-приложений получение HTML-документа естественно, поскольку это их основная выходная форма. Важно также, что HTML (точнее, его синтаксически корректная версия XHTML) – XML-язык. Получить HTML-документ напрямую программным компонентом ИС проще и быстрее, но если его предполагается макетировать в нескольких форматах, то мы применяем описываемую многостадийную схему, чтобы воспользоваться ее преимуществами. В этом случае выполняются (универсальные для всех форм) первые две подпрограммы ПГД, а для макетирования используются XSL-преобразования (XSLT) и стилевые таблицы CSS.

MS Word. Учитывая популярность текстового процессора Word, целесообразно генерировать Word-документы во многих задачах, в которых не требуется высококачественная полиграфия (формирование отчетных и аналитических документов, представление информационных выборок и т. д.). Предпосылки: а) современные версии MS Word используют для внутреннего представления документов XML-язык WordML (WML); б) Word может взаимодействовать с web-приложением посредством ActiveX; в) Word имеет встроенный XSL-процессор.

Запущенная пользователем ПГД реализует описанную схему (см. рисунок), формируя на стадии макетирования WML-документ соответствующим XSLT, затем запускает на компьютере клиента MS Word посредством ActiveX и передает ему полученный WML-документ. WML-документ самодостаточен, поскольку WML – язык внутреннего представления документа Word: на нем описываются все элементы документа, стилевые параметры, динамические компоненты (оглавление, колонтитулы, нумерация страниц и др.).

Таким образом, в результате запуска ПГД пользователь получает сгенерированный и отмакетированный документ непосредственно открытым в Word и готовым к прочтению, редактированию, записи и печати.

Corel Ventura (CV). Если от ИС требуется формировать оригинал-макеты для полиграфической публикации, Word уже не подходит – для этой задачи мы разработали схему подключения издательской системы Corel Ventura.

Работа с CV сложнее: нет возможности сформировать самодостаточный XML-документ, набор методов ActiveX-взаимодействия с CV минимален, что не позволяет web-приложению выполнить команды CV для макетирования. Наше know how – выполнить сценарии CV, а если нам нужно передать им данные, то поместить их во входном XML-документе в виде меток CV, невидимых при отображении документа (например, так мы отмечаем места и параметры сложных объединений ячеек таблиц) [4]. ПГД выполняет типовую последовательность действий, запускает CV с шаблоном документа (которые содержат нужные сценарии) и передает ему сформированный XML-файл и VMF-файл, в котором указываются параметры преобразования XML-элементов в компоненты CV-документа. Сценарии завершают макетирование, формируя динамические структуры и выполняя другие доработки. Пользователь получает открытый в CV документ, готовый к дальнейшим действиям с ним.

LaTeX. Несколько лет назад в НИЛИТС был создан модуль ITSLTeX, который «научил» LaTeX «понимать» синтаксис XML [1, 2]: он способен воспринимать XML-файл в качестве входного, преобразовывать XML-теги в команды и макроопределения LaTeX и компилировать документ в форматы PostSript или PDF. Пока мы не добивались «горячего» подключения ITSLTeX к web-приложению, но предполагаем, что для определенных приложений заманчива возможность макетировать на сервере документ средствами ITSLTeX и передавать его пользователю в PDF-формате.

Автоматическая генерация компонентов ПГД

Одна из концептуальных идей нашей технологии – поиск оптимального разделения декларативной и процедурной составляющих знаний и организация эффективного их взаимодействия. Мы стремимся выделить и «вынести за скобку» общие группы знаний, чтобы обеспечить их многократное использование и получить другие преимущества: минимизацию проектных затрат, унификацию и нормализацию форм, единственность декларирования и др. [5]. XML-технология дает необходимые для этого возможности, позволяя формулировать на XML-языках обе составляющие знаний. Покажем это на примере.

С помощью разных приложений можно получить на бумаге одинаковые образы электронного документа. Есть различия в моделях представления документа в разных приложениях и форматах, но на бумаге это уже несущественно. С одной стороны, мы можем описать в единых терминах общую картину планарной формы генерируемых документов, создав *полиграфическую спецификацию*. С другой стороны, мы обладаем знаниями, как отразить эту спецификацию в конкретном формате или приложении: мы выражаем ее в XSLT, CSS, VMF, макроопределениях LaTeX как декларативно (CSS, VMF), так и процедурно.

Пройдя этап «ручного» формулирования этого блока знаний и апробировав технологические цепочки, мы переходим к выделению общего и эффективному разделению знаний. В данном случае:

– создается XML-язык полиграфических спецификаций классов виртуальных документов, предусмотренных в данной ИС (или более универсальный);

– формулируются процедурные «метазнания», которые способны трансформировать описанную на этом языке спецификацию в компоненты ПГД-МП (XSLT, CSS, VMF и др.); для этого вполне подходит XSLT (т. е. XSLT, представляющая более общие знания, порождает XSLT, представляющую более конкретные знания, которая, в свою очередь, из более общей XML-формы порождает более конкретную XML-форму документа);

– для конкретных классов виртуальных документов описывается полиграфическая спецификация (декларативная составляющая знания), а компоненты ПГД-МП (как декларативные, так и процедурные) генерируются с помощью разработанных процедурных компонентов.

Полученные результаты исследований, экспериментов и практических разработок позволяют наделить web-системы способностью компоновать собранную по запросу пользователя информацию в качественные по форме документы, создавать их в различных форматах и передавать для воспроизведения и последующей обработки в различные приложения (текстовые процессоры и издательские системы).

Понятно, что еще проще генерировать документы более жесткой структуры, подключая более широкий круг приложений. Например, многие web-системы предлагают пользователю скачать с сервера файл MS Excel, содержащий массив структурированной информации. Эти файлы формируются заранее и хранятся на сервере. Если даже документы качественны по форме, такой подход не соответствует требованиям полноты и точности удовлетворения ИПП: а) нужная пользователю информация может содержаться в нескольких файлах, б) эти файлы могут нести множество лишней информации,

а ознакомиться с их содержанием без перекачки с сервера нельзя. Применив наш подход, можно существенно улучшить эксплуатационные характеристики ИС – динамически формировать документы, полно и точно соответствующие запросу пользователя, и представлять их в формах электронных таблиц, диаграмм, графиков и др., подключив соответствующие ActiveX-компоненты и приложения. С его помощью несложно обеспечить и генерацию HTML-оглавлений, аннотаций, других компактных форм, что дает пользователю возможность ознакомиться с содержанием крупного документа, прежде чем он будет сгенерирован и передан по сети. В последние годы XML получает все большее применение в различных приложениях, совершенствуются и расширяются способы взаимодействия приложений, в том числе с web-приложениями, стандартизируются и унифицируются формы документов и языки разметки. Это позволяет оценивать предлагаемый подход как находящийся в русле мировых тенденций и соответствующий перспективным направлениям развития ИС.

Важный вектор развития XML-технологии публикации лежит в плоскости инженерии знаний. Подход, основанный на разделении и представлении в XML-форме декларативной и процедурной составляющих знаний, замечателен производимым кумулятивным эффектом: выделяя инварианты знаний и затем соединяя их в различных комбинациях, можно добиться быстрого роста функциональности и гибкости ИС при минимальных проектных затратах. Здесь важно отметить, что после выделения блока декларативных данных средствами ИС нетрудно обеспечить его «ведение» не программистами этой ИС, а специалистами других профилей – инженерами знаний, экспертами, дизайнерами и др. Мы показали, что полиграфическую спецификацию класса документов можно вынести как декларативный компонент, обеспечив затем автоматический перевод этих знаний в подпрограммы макетирования для различных форматов и приложений. Методически это означает, что программисты закончили свою работу, теперь многообразие полиграфических спецификаций могут создавать другие специалисты вплоть до того, что функцией управления внешним видом документа можно наделить конечного пользователя ИС.

Аналогичными соображениями продиктована и трехуровневость технологической схемы генерации документов: она разработана таким образом, чтобы при необходимости подключения других приложений для передачи в них документов понадобились лишь проектные усилия, нужные для учета специфики приложения, а остальная часть схемы могла бы использоваться в неизменном виде. Помимо минимизации проектных затрат это обеспечивает также унификацию представления документов. Понятно также, что варианты трех подпрограмм ПГД могут соединяться в различных комбинациях, обеспечивая многообразие форм документов.

В целом, апробировав описанные в данной статье решения и проанализировав перспективные направления их развития, мы можем сделать вывод о том, что они могут иметь широкий спектр применения и способствовать достижению нового уровня функциональных возможностей и качественных характеристик Интернет- и интранет-систем, а также сокращению затрат на проектирование таких систем и повышению качества проектирования.

1. Король И.А., Чеушев В.А., Вяжевич В.Л., Орлов В.А. // Материалы I Международной конференции «Информационные системы и технологии»: в 2 ч., Минск, 5–8 нояб. 2002 г. Мн., 2002. Ч. 1. С. 54.

2. Курбацкий А.Н., Чеушев В.А., Радиванович Н.Н. и др. // Материалы международной научно-практической конференции «Современные компьютерные технологии в системах правовой информации», Минск, 21–22 нояб. 2002 г. Мн., 2002. С. 211.

3. Курбацкий А.Н., Чеушев В.А., Радиванович Н.Н. и др. // Проблемы правовой информатизации: Науч.-практ. журн. Мн., 2002. Вып. 5. С. 101.

4. Курбацкий А.Н., Чеушев В.А. // Материалы II Научно-практической конференции «Управление информационными ресурсами», Минск, 16 марта 2004 г. Мн., 2004. С. 22.

5. Курбацкий А.Н., Чеушев В.А. // Материалы научно-практической конференции «Управление информационными ресурсами», Минск, 15 мая 2003 г. Мн., 2003. С. 63.

6. Чеушев В.А. // Проблемы правовой информатизации. Мн., 2004. № 2(8). С. 29.

Поступила в редакцию 04.03.08.

Александр Николаевич Курбацкий – доктор технических наук, профессор, заведующий кафедрой технологии программирования.

Василий Александрович Чеушев – кандидат технических наук, заведующий лабораторией Центра информационных ресурсов и коммуникаций БГУ.

Бинь Сюе – аспирант кафедры технологии программирования. Научный руководитель – А.Н. Курбацкий.

Светлана Валентиновна Гурновская – инженер-программист Центра информационных ресурсов и коммуникаций БГУ.