

УДК 004.91

В. В. МИРОНОВ, Г. Р. ШАКИРОВА

ПРОГРАММНО-ИНСТРУМЕНТАЛЬНОЕ СРЕДСТВО ДЛЯ СОЗДАНИЯ И ВЕДЕНИЯ ДИНАМИЧЕСКИХ XML-ДОКУМЕНТОВ

Обсуждаются вопросы практического приложения XML-технологий к концепции динамических документов. Формулируется концепция XML-документов со встроенной динамической моделью, приводится общая структурная схема этой модели и описывается метод ее интерпретации. Обсуждается архитектура разработанного программно-инструментального средства создания и ведения динамических XML-документов, приводится краткий обзор составляющих его модулей. *Динамический документ; встроенная динамическая модель; XML-технологии; динамический XML-документ; программно-инструментальное средство*

ВВЕДЕНИЕ

В традиционных системах электронного документооборота чаще всего имеет место ситуация, когда модель создания и применения документа отделена от него самого. В результате документы сначала создаются, а затем осуществляется их привязка к некоторой модели. Сам по себе этот процесс достаточно трудоемкий, кроме того, полученный результат может не в полной мере отвечать поставленной задаче, что обусловлено ошибками, связанными с человеческим фактором.

В связи с этим в работе [1] была предложена концепция динамических документов. Под динамическим документом (ДД) понимается документ, который содержит встроенную динамическую модель своего создания и использования [1]. Концепция динамических документов была реализована на базе текстового процессора Microsoft Word. Динамические документы были представлены в виде Word-документов, а для их интерпретации использовались функциональные возможности текстового процессора и языка программирования VBA. Динамическая модель была реализована в виде совокупности макросов, каждый из которых соответствует ее определенному элементу.

Безусловно, подобная организация ДД обладает массой преимуществ. В первую очередь, это повсеместная распространенность и огромная функциональность Microsoft Word. Кроме того, унифицированный графический интерфейс пользователя дает возможность свести к минимуму затраты на освоение этого программного продукта.

Однако у данного подхода есть и недостатки. Основным из них является низкая пригодность для публикации информации в Интернете. Для этого документы конвертируются процессором MS Word в HTML-файлы, перегруженные специфической информацией редактора, которая в значительной степени усложняет работу с результирующим документом. Среди других недостатков можно отметить низкий уровень межплатформенной переносимости, сложность внутреннего представления динамической модели и т. д.

Для устранения вышеописанных недостатков предлагается представлять динамические документы в XML-формате и применять для их обработки программные средства, использующие сопутствующие XML-технологии.

На сегодняшний день XML является самым популярным форматом представления данных. Это связано с тем, что XML обладает рядом преимуществ перед другими форматами. Среди них можно отметить независимость от платформ, поддержку производителями ПО, расширяемость, отделение данных от представления и т. д.

С точки зрения концепции ДД, по сравнению с Word-форматом применение XML-разметки обладает следующими преимуществами:

- формат XML-документа, характеризующийся древовидным представлением структуры его элементов, оптимально отражает иерархическую структуру встроенной модели динамического документа;

- объектность. Каждый тег XML-документа может быть поставлен в соответствие элементу динамической модели. Наряду со свойством иерархичности это позволяет наиболее информативно сформировать структуру внутреннего представления встроеной модели динамического документа;

- XML позволяет ввести унифицированную структуру динамических документов. Для этого используется:

- 1) пространство имен с зарезервированным именем, определяющее имена специальных типов элементов динамического XML-документа и их атрибутов;

- 2) XML-схема, задающая структуру специальных типов элементов динамического XML-документа и их атрибутов.

- свойство независимости от платформы, присущее XML-технологии, позволяет обеспечить переносимость динамических документов на их основе, что немаловажно при обмене данными между приложениями и платформами.

Перечисленные преимущества делают XML оптимальной технологией для создания и использования динамических документов со встроеной динамической моделью по сравнению с подходом, основанным на механизмах текстового процессора MS Word. В связи с этим в работах [2, 3] авторами было введено понятие динамических XML-документов, основанное на применении XML-технологий к концепции ДД.

Целью настоящей статьи является обсуждение вопросов, связанных с практической реализацией как самих динамических XML-документов, так и механизмов их создания и ведения.

1. КОНЦЕПЦИЯ ДИНАМИЧЕСКИХ XML-ДОКУМЕНТОВ

В основе понятия динамического XML-документа (XML-DD) лежит концепция ДД. В электронный документ встраивается динамическая модель, отражающая процессы создания и использования документа. Эта модель задает совокупность состояний (ситуаций) соответствующего процесса, а также набор переходов между ними.

В каждый момент времени модель находится в одном или нескольких состояниях. Для их указания применяется память текущего состояния (ПТС), в которой отражается смена состояний модели.

Содержательная часть исходного электронного документа в ДД представляется в виде прикладных фрагментов, ассоциированных с ситуациями встроеной динамической модели.

Таким образом, ДД представлен двумя составляющими [1]:

- содержательная база документа, представляющая собой упорядоченную совокупность фрагментов информации (текстовых, графических, мультимедийных и т. п.);

- динамическая модель документа, которая, в свою очередь, включает две составляющие:

- исходная динамическая модель, которая создается разработчиком структуры документа на подготовительном этапе;

- модель текущего состояния, которая строится для конкретного документа в ходе его жизненного цикла.

Особенность концепции XML-DD заключается в том, что все компоненты ДД представляются в форме XML-фрагментов.

При этом используемая в XML-DD разметка не является произвольной, а регламентируется общей структурной моделью документа. В работе [2] авторами выделены два типа разметки:

- прикладная, представленная тегами, которые должны структурировать документ независимо от текущего состояния;

- ситуационная, включающая теги, определяющие информацию, относящуюся к различным состояниям.

Прикладная разметка используется для введения фрагментов документа, ассоциированных с ситуациями. Она формируется на этапе создания документа, после или параллельно с построением исходной динамической модели.

Для обеих составляющих встроеной динамической модели применяется только ситуационная разметка, которая задается на этапе построения (исходная динамическая модель) или применения (ПТС) документа.

В самом жизненном цикле XML-DD выделяются те же этапы, что и для ДД. Во-первых, построение этих документов, когда разработчик определяет структуру встроеной модели, а также задает соответствие между ее элементами и прикладными фрагментами. На втором этапе осуществляется уже непосредственная работа с XML-DD. Пользователю предоставляется инструментарий навигации по исходной динамической модели, ис-

пользуя который он обращается к фрагментам, ассоциированным с определенными ситуациями.

Разметка XML-DD является самостоятельным языком описания данных, разработанным в соответствии с синтаксисом XML. Ее обработка должна базироваться на функциональных возможностях XML-парсера, осуществляющего проверку синтаксиса документа. Однако для выполнения логических операций, связанных с обработкой встроеной модели, стандартных средств недостаточно. Поэтому необходим внешний интерпретатор, который должен сочетать в себе механизмы XML-процессора и обработчика встроеной модели, т. е. выполняющий обработку документов типа XML-DD как с точки зрения их синтаксиса, так и семантики.

2. СТРУКТУРНАЯ МОДЕЛЬ ДИНАМИЧЕСКОГО XML-ДОКУМЕНТА

Динамическая модель (ДМ) определяет совокупность состояний, в которых могут находиться процессы создания и использования документа, переходы между ними, а также условия срабатывания этих переходов и выполняемые при этом действия. Она относится к классу дискретно-событийных моделей и может быть представлена в форме сетей Петри, потоковых диаграмм, графов переходов, конечных автоматов и т. д. В работе [1] предлагается реализовывать ДМ в классе иерархических ситуационных моделей (ИСМ).

Синтаксис ИСМ задает четыре группы объектов: модель, подмодель, состояние, переход. Модель представлена подмоделями, объединяющими группы состояний, связанных друг с другом переходами и содержащих погружения в другие дочерние подмодели.

В составе ДМ эти объекты определяют структуру исходной модели. Кроме того, в соответствии с концепциями ДД и XML-DD, добавлены дополнительные элементы, соответствующие ассоциированным фрагментам и принадлежащие объектам-состояниям.

В XML-DD прикладные фрагменты интегрированы в саму динамическую модель, а не хранятся отдельно, как в ДД, представленных в работе [1]. Внутренние предикаты также выделены в отдельные элементы переходов, поскольку для некоторых переходов может потребоваться выполнение нескольких условий.

Согласно концепции XML-DD, ДМ представлена в XML-формате. Объектность XML-разметки позволяет каждому объекту модели поставить в соответствие XML-элемент, а

его свойства указать в соответствующих атрибутах. В зависимости от типа объекта модели, соответствующие XML-элементы содержат определенный набор свойств и вложенных элементов. Свойства объектов модели (например, название) задаются с помощью атрибутов соответствующих XML-элементов.

Для нового типа элемента — предиката перехода — заданы два дополнительных свойства: тип и состояние-параметр. В XML-DD внутренние предикаты относятся к одному из двух типов. Оба этих типа проверяют статус состояния-параметра, только один проверяет, является ли данное состояние текущим, а другой — наоборот.

Еще один новый тип элемента — прикладной фрагмент («Ref»), ассоциированный с ситуацией. Он содержит стандартный атрибут, в котором указывается имя элемента. Сам фрагмент хранится в форме текстового содержимого узла Ref. С целью обеспечения переносимости и гибкости функционирования каждый фрагмент задается также в XML-формате, а именно, с помощью разметки WordML. Чтобы отделить теги, составляющие содержательную часть фрагмента, от соответствующего элемента модели, используется формат CDATA. CDATA — специальная форма задания текстовых значений, которая допускает использование зарезервированных символов [3]. С каждым состоянием может быть связано несколько прикладных фрагментов.

Для ПТС вводится одноименный XML-элемент, в который по мере использования документа добавляются элементы, соответствующие текущим состояниям. В его структуре представлены два типа элементов — подмодели и состояния. Переход, в результате выполнения которого некоторое состояние стало текущим, указывается в качестве атрибута «jump_source» соответствующего элемента. Для каждого элемента фиксируется дата и время занесения в ПТС, а также XPath-выражение соответствующего элемента исходной модели.

Для отделения ситуационной разметки от прикладной введено пространство имен с префиксом *hsm*, задающее именованное множество допустимых имен элементов и атрибутов.

XML-технологии позволяют описать структуру и содержимое XML-DD, тем самым создать универсальный механизм, регламентирующий структуру документа. Предлагается использовать схему XML-документа.

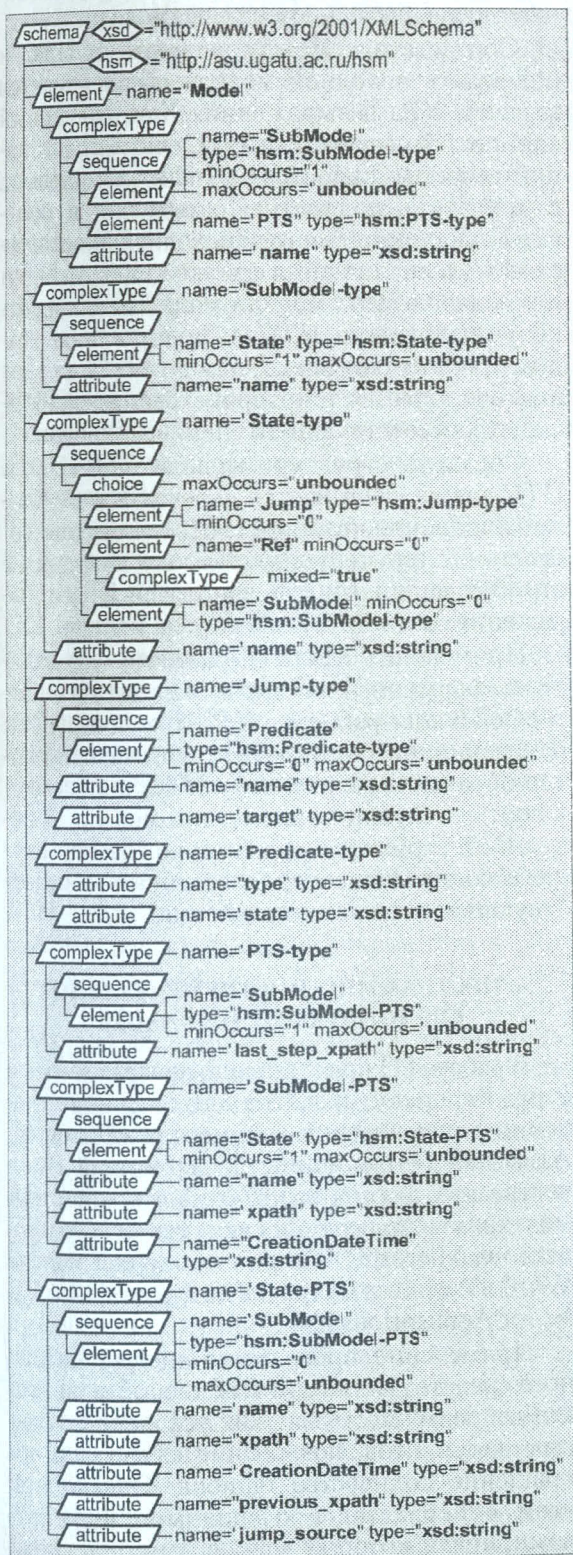


Рис. 1. Концептуальная модель XSD-схемы динамического XML-документа

В общем случае схема документа — это модель, отделенная от самого документа, в которой заданы его структурные и параметрические ограничения. Она определяет, что должно и чего не должно быть в документе. Существует три вида схем — DTD, XDR и XSD, и только одна из них — XSD — утверждена консорциумом W3C. Именно ее предлагается использовать для XML-DD.

В схеме для каждого XML-элемента динамической модели указывается его тип и допустимое число экземпляров, задаются вложенные элементы и атрибуты, для каждого из которых определяется тип данных и способ использования (рис. 1).

3. МЕТОД ИНТЕРПРЕТАЦИИ ДИНАМИЧЕСКОГО XML-ДОКУМЕНТА

Интерпретация модели ДД основана на отслеживании изменений ее текущих состояний. При этом обработка модели ДД является асинхронной, т. е. смена текущего состояния происходит в результате наступления некоторого события. Согласно структуре модели ДД, состояния одного уровня иерархии связаны друг с другом переходами. Именно выполнение этих переходов и является тем событием, которое ведет к смене текущего состояния.

Выполнение переходов, в свою очередь, также связано с наступлением событий. Это событие — выполнение предиката, связанного с данным переходом. Это может быть либо внешний, либо внутренний предикаты, либо их комбинация. Внешний предикат выполняется в соответствии с выбором пользовательского инструментария. Внутренний предикат имеет дело с внутренними состояниями модели, причем проверяемые состояния являются дочерними (однако необязательно непосредственными потомками) по отношению к состоянию, из которого выполняется переход.

Однако переходы затрагивают изменения текущего состояния только на одном уровне иерархии. В то же время состояния могут содержать внутренние подмодели, переход к которым называется погружением. Предлагается сделать подобные погружения автоматическими, т. е. как только некоторое состояние становится текущим, сразу выполняются все его погружения. При этом текущими становятся начальные состояния соответствующих подмоделей. При этом само родительское состояние не перестает быть текущим. В подмодели начинается собственный процесс изменения текущего состояния, который заверша-

ется, когда родительское состояние перестает быть текущим. Поскольку обработка является асинхронной, то погружения выполняются в том порядке, в котором они заданы в модели. Как правило, в качестве параметров внутренних предикатов переходов из такого состояния выступают состояния этих подмоделей, подмоделей следующего уровня иерархии и т. д. Поэтому если некоторое состояние не содержит подмоделей, то и переходы из него не содержат внутренних предикатов.

Таким образом, текущим может являться не одно состояние, а их совокупность с учетом всех текущих состояний в подмоделях данного текущего состояния, т. е. формирование ПТС происходит по двум контурам: всей модели и подмоделей текущего состояния.

Еще одной задачей интерпретации является организация диалогового взаимодействия между пользователем и документом, а точнее, встроенной моделью этого документа. Для этого строится навигационный механизм, полностью определенный содержимым ПТС.

Особенность интерпретации XML-DD заключается в том, что все объекты встроенной динамической модели являются XML-элементами. Это позволяет применить в процессе обработки модели XPath-адресацию.

Каждый элемент модели уникальным образом характеризуется своим собственным XPath-выражением, указывающим его местоположение в иерархии остальных элементов. Это позволяет не вводить каких-либо идентификаторов элементов модели, а использовать их XPath-выражения.

XPath-выражения применяются в качестве параметров переходов, однозначно определяя состояние, к которому должно привести выполнение данного перехода. В качестве параметров предикатов также целесообразно использовать XPath-выражения, содержащие путь к соответствующему состоянию относительно состояния-источника.

Еще один аспект применения XPath-выражений связан с формированием ПТС. Согласно структуре модели XML-DD, в состав ПТС входят подмодели и состояния. На начальном этапе в ПТС вносятся элементы, соответствующие подмоделям верхнего уровня и их начальным состояниям. Последующие изменения ПТС сопровождаются внесением новых состояний в соответствующие подмодели, а также создание новых элементов-подмоделей. При этом каждое новое текущее состояние не удаляет из ПТС прежнее текущее

состояние, а просто сдвигает его вниз. Таким образом, текущим является то состояние, которое в ПТС является первым в иерархии соответствующего элемента-подмодели.

Соответствие между элементами ПТС и объектами исходной модели обеспечивается за счет XPath-выражений. Каждый новый элемент, занесенный в ПТС, является, с одной стороны, самостоятельным элементом, а с другой — полностью определяется соответствующим элементом исходной динамической модели, является ссылкой на него. Один и тот же элемент модели может неоднократно участвовать в ПТС в форме таких ссылок. Для этого в состав элемента ПТС входит атрибут «xpath», в котором хранится XPath-адрес соответствующего элемента модели.

Согласно структуре модели XML-DD, в ПТС также указывается переход, в результате выполнения которого соответствующее состояние стало текущим. Для его хранения в атрибут «jump_source» заносится XPath-выражение соответствующего узла модели.

Применение XPath-выражений определяет и формирование пользовательского интерфейса для работы с ДД. При построении навигационного инструментария интерпретатор обращается к элементу ПТС, а от него — к соответствующим элементам модели. В результате структура модели в контексте текущего состояния воспроизводится в соответствующем элементе управления.

4. ПРОГРАММНО-ИНСТРУМЕНТАЛЬНОЕ СРЕДСТВО XML-DD MANAGER

В работе [1] представлено программно-инструментальное средство для создания и ведения динамических документов «Dissertant Assistant» («Помощник диссертанта»). В качестве платформы разработки этого инструментария используется среда текстового процессора Microsoft Word — объектная модель Word и средства макросов на языке Visual Basic for Applications [1].

Применение программно-инструментального средства Dissertant Assistant обладает такими преимуществами, как ускорение создания документов, повышение качества и сокращение количества ошибок. Однако при этом он обладает и очевидными недостатками, среди которых можно отметить сложность публикации документов в Интернете, низкую межплатформенную переносимость и т. д. Кроме того, он не является универсальным, поскольку жестко привязан к той практической задаче, которая ставилась перед

разработчиком (а именно, к созданию и ведению комплекса документов из диссертационного дела соискателя ученой степени). Изменение контекста или назначения динамических документов потребует создания нового программно-инструментального комплекса, ориентированного уже на другую задачу. Кроме того, рассматриваемый комплекс не ориентирован на XML-формат динамических документов. Конечно, текстовый процессор Word позволяет работать с XML-документами, для этого он даже поддерживает специальный формат – WordML. Однако его применение не всегда оправданно. Причина – та же, что и при публикации документов в Интернете: добавление в исходный документ большого набора служебной информации, что затрудняет его просмотр и последующую интерпретацию.

Открытость XML-формата позволяет выполнять их просмотр и редактирование с помощью стандартных текстовых редакторов (MS Notepad, WordPad и т.д.). При этом нет никакой гарантии соблюдения, по меньшей мере, структуры встроенной модели, потому что текстовые редакторы позволяют бесконтрольно вносить любые изменения в исходный документ, особо не вдаваясь в его семантику.

Использование существующих XML-редакторов также неэффективно. Очевидно, что XML-редакторы позволяют выполнять просмотр и редактирование документов в рамках соответствующей XSD-схемы. Это позволяет строго соблюдать структуру исходного XML-DD-документа. Но при этом правильность изменений может быть проверена только после их внесения. Существующий механизм редактирования рассматривает каждый компонент документа как обычный XML-элемент или атрибут, без учета его семантики. Это существенно затрудняет работу пользователя с динамическими XML-документами. Кроме того, работа с динамическим XML-документом не ограничивается только его редактированием. Такие операции, как навигация по встроенной модели, работа с ассоциированными фрагментами, выполнить в рамках стандартных XML-редакторов невозможно.

Таким образом, необходим такой инструментарий, который реализовывал бы универсальные механизмы любых манипуляций с XML-документами со встроенной динамической моделью.

В результате было разработано программно-инструментальное средство создания и ведения динамических XML-документов «XML-DD Manager». В нем учтена специфика документов, отвечающих концепции XML-DD, и разработаны механизмы, реализующие метод интерпретации структурной модели XML-DD.

В этом программно-инструментальном средстве были выделены 4 программных модуля:

- основной модуль;
- модуль навигации;
- модуль редактирования;
- модуль работы с ассоциированными фрагментами.

Каждый из этих модулей имеет сложную внутреннюю структуру, представленную набором функциональных подмодулей, отвечающих за реализацию подзадач модуля. Обобщенная структура разработанного программно-инструментального средства представлена на рис. 2.

Основной модуль является связующим звеном между всеми остальными модулями программно-инструментального средства. Именно этому модулю автоматически передается управление при загрузке программы. В рамках основного модуля выполняются только две операции:

- вызов всех остальных программных модулей;
- задания пользовательских настроек, необходимых для последующей работы с XML-DD (путь к исходному файлу, графические настройки и т.д.).

Модуль навигации. Модуль навигации позволяет пользователю перемещаться между ситуациями в динамической модели. Организация навигации по встроенной модели XML-DD выполняется в два этапа: заполнение памяти текущего состояния и формирование на ее основе навигационного инструментария. Поэтому на верхнем уровне в модуле навигации были выделены два подмодуля:

- модуль работы с ПТС;
- модуль создания пользовательского интерфейса навигации.

Первый модуль, как следует из названия, отвечает за отслеживание изменений в ПТС, что является основой работы всего модуля навигации. ПТС динамически обновляется по мере смены текущего состояния при активации некоторого перехода по выбору пользователя или при срабатывании соответствующего предиката.

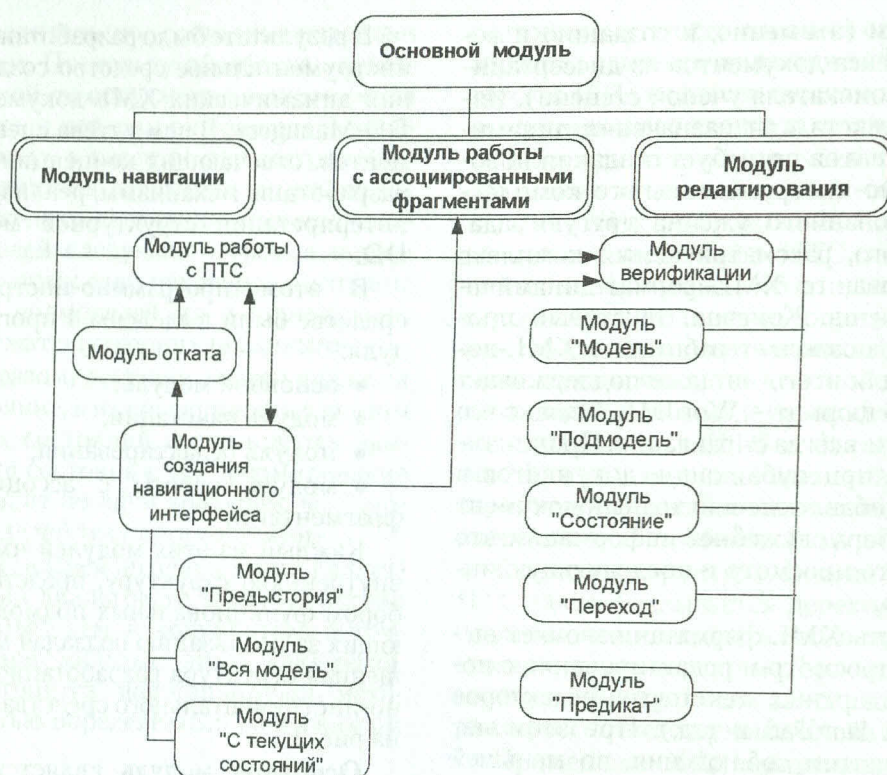


Рис. 2. Обобщенная архитектура программно-инструментального средства «XML-DD Manager 1.0»

Второй модуль, основываясь на содержимом ПТС, формирует соответствующий инструментарий навигации. Для этого создается навигационный механизм, в котором представлена вся иерархия объектов встроенной модели XML-DD. Главная цель навигационного инструментария состоит в том, чтобы отобразить структуру встроенной модели так, чтобы пользователь без труда мог перемещаться по ее элементам с учетом их специфики. Поэтому для представления встроенной модели в рамках пользовательского инструментария предлагается взять класс TreeView (дерево). Этот класс предназначен для отображения иерархических структур, к которым относится и встроенная модель, являющаяся основой для построения механизма навигации по XML-DD. Таким образом, в качестве навигационного инструментария вводится так называемое *дерево навигации*. Для представления XML-элементов в рамках этого дерева навигации был разработан новый класс XMLTreeNode, базирующийся на классе TreeNode.

Навигационный механизм доступен в трех режимах представления, каждому из которых соответствует свой программный модуль:

- Режим «С текущих состояний». Этот режим загружается по умолчанию. Здесь доступна уже не вся модель, а та ее часть (подмодель), которая берет свое начало от некоторой текущей ситуации, причем сразу несколько таких «частей» модели могут быть активны одновременно. Еще одной особенностью рассматриваемого режима навигации является то, что помимо допустимого перехода из текущего состояния доступна для просмотра и иерархия объектов модели, в вершине которой находится состояние, являющееся целью перехода (рис. 3¹).

- Режим «Вся модель». Здесь доступна для просмотра вся встроенная модель, а для каждого состояния показаны возможные переходы и погружения. Текущие состояния просто особым образом выделяются в контексте всей иерархии объектов.

¹Приведенные в статье примеры иллюстрируют динамический XML-документ, в основе которого лежит текст «Федерального закона о несостоятельности (банкротстве)», рассмотренный авторами в работе [2]

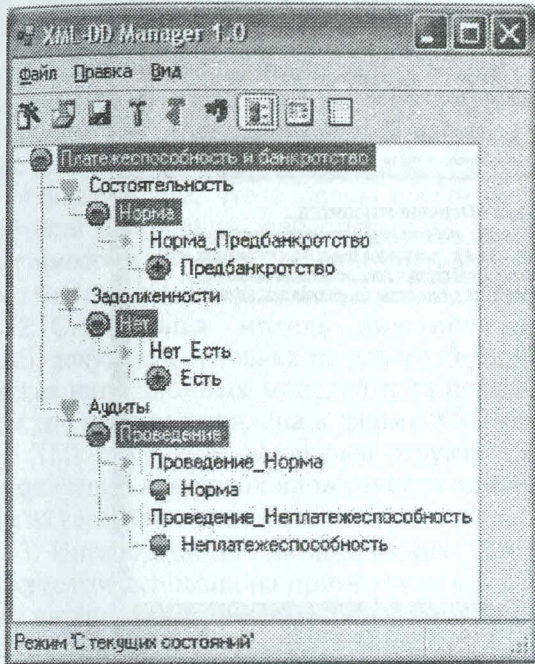


Рис. 3. Навигационный механизм динамического XML-документа. Режим «С текущих состояний»

• Режим «Предыстория». В этом режиме пользователь может просмотреть предысторию текущего состояния. При этом формируется последовательность «подмодель → состояние → переход → состояние» выполненных шагов навигации пользователя.

Любое изменение текущей ситуации можно отменить с помощью функции отката. Он приводит ПТС к тому состоянию, в котором она находилась вплоть до выполнения последнего перехода. Для реализации этой функции был разработан соответствующий программный модуль.

Модуль редактирования. Модуль редактирования предназначен для выполнения функций создания и изменения структуры встроенной модели XML-DD. Он оперирует инструментарием редактирования (деревом редактирования), который отражает всю совокупность элементов динамической модели XML-DD без привязки к текущему состоянию.

Каждому типу объекта модели соответствуют свои операции редактирования. Поэтому при выборе пользователем объекта соответствующие классы модуля редактирования определяют его тип, и в соответствии с ним уже другие классы формируют список доступных действий редактирования. Модуль редактирования предусматривает определение пяти типов объектов — модель, подмодель, состояние, переход, предикат. каждо-

му из них соответствуют одноименные подмодули редактирования.

Например, для элемента типа «Состояние» (рис. 4) предусмотрены следующие операции редактирования:

- «Новое состояние». Эта функция предназначена для создания нового элемента типа «Состояние»;
- «Удалить состояние». В результате выполнения этой операции выбранное состояние будет удалено из исходной модели XML-DD;
- «Переименовать». Эта функция позволяет переименовать выбранное состояние, т. е. изменить значение атрибута *name* соответствующего XML-элемента;
- «Сделать состояние начальным». Согласно модели XML-DD, порядок следования состояний в иерархии не имеет значения. Исключением является только начальное состояние. Выполнение этой функции позволит перевести выбранное состояние в статус начального;
- «Создать» → «Подмодель» → «Переход». Эта функция предназначена для создания нового дочернего элемента выбранного состояния, относящегося к типу «Подмодель» или «Переход» соответственно.

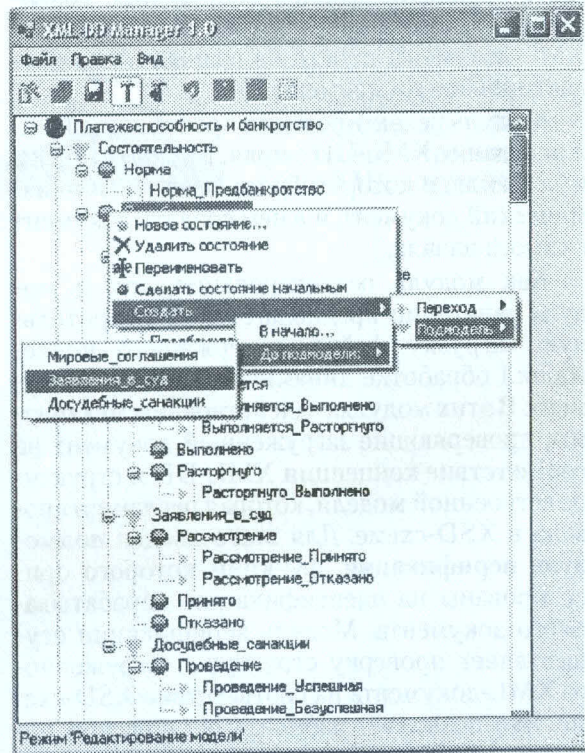


Рис. 4. Механизм редактирования модели динамического XML-документа

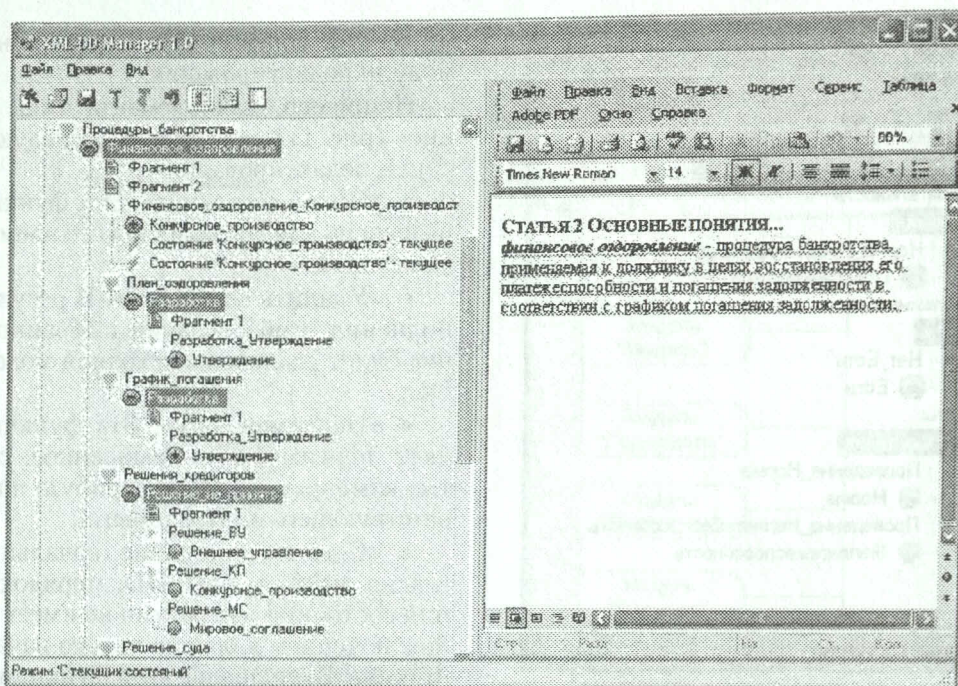


Рис. 5. Пример фрагмента документа, ассоциированного с текущей ситуацией

Все выполненные в дереве редактирования изменения отражаются в исходном файле XML-DD. При этом на любом этапе редактирования или смены текущего состояния можно сохранять XML-DD в файл, отличный от исходного. Теоретически можно создать несколько файлов сохранения, содержащих по сути один и тот же динамический XML-документ, просто на разных этапах редактирования и применения.

Модуль редактирования предполагает также создание XML-DD с нуля. Можно выбрать имя файла, в котором будет храниться динамический документ, и в нем создать документ с самого начала.

Как модуль редактирования, так и модуль навигации предполагают предварительную загрузку файла, содержащего подлежащий обработке динамический XML-документ. В этих модулях задействованы механизмы, проверяющие загруженный документ на соответствие концепции XML-DD и структуре встроенной модели, которая регламентирована в XSD-схеме. Для этого введен **подмодуль верификации**, функции которого ориентированы на идентификацию обрабатываемого документа. Модуль верификации осуществляет проверку структуры загруженного XML-документа на соответствие XSD-схеме. Это позволяет избежать ошибок при попытке интерпретации документа неопределенного формата с помощью методов, пригод-

ных исключительно для XML-документов со встроенной динамической моделью.

Модуль работы с ассоциированными фрагментами. Согласно концепции XML-DD, с каждым состоянием встроенной модели ассоциированы один или несколько фрагментов документа. Поэтому в «XML-DD Manager» предусмотрена возможность последовательной обработки каждого фрагмента, ассоциированного с той или иной ситуацией.

Особенностью рассматриваемого программного модуля является то, что он обеспечивает интеграцию приложения Microsoft Word 2003 для оперирования ассоциированными фрагментами.

Для хранения ассоциированных фрагментов используется формат WordML. Модуль работы с текстом позволяет добавлять новые или изменять уже существующие WordML-фрагменты. Модуль запускает текстовый процессор Microsoft Word 2003, поддерживающий работу с документами в формате WordML.

Особенностью работы этого модуля является то, что текстовый процессор запускается не в отдельном окне, а внутри основной формы инструментария (рис. 6). Все выполненные во фрагменте изменения, в том числе операции форматирования, вставки объектов, снова записываются в исходный файл с XML-DD в формате WordML.

ВЫВОДЫ

1. Концепция динамических XML-документов основана на встраивании в документ динамической модели, с состояниями которой ассоциированы прикладные фрагменты, и отличается тем, что с целью обеспечения гибкости построения, переносимости и совместимости динамический документ строится в классе XML-документов

2. Структурная модель динамического XML-документа основана на классе иерархических ситуационных моделей и отличается тем, что она представлена в форме XSD-схемы XML-документа, задающей структурные ограничения и порядок вложенности элементов на введенном пространстве имен.

3. Интерпретация динамических XML-документов основана на принципе асинхронной интерпретации и отличается тем, что для однозначной идентификации элементов модели используется механизм XPath-адресации, что позволяет установить однозначное соответствие между элементами модели и ПТС.

4. Программно-инструментальное средство «XML-DD Manager» основано на XML-технологиях и принципах объектно-ориентированного программирования и отличается тем, что для реализации предложенных концепции, модели и метода и обеспечения универсальности используется новый класс на базе стандартного класса TreeView, который обеспечивает функциональность по созданию, редактированию и навигации по динамической модели и администрированию информационного наполнения документа в контексте текущего состояния.

СПИСОК ЛИТЕРАТУРЫ

1. **Гарифуллин, Т. А.** Обеспечение целостности комплекса электронных документов на основе встраиваемых динамических моделей : автореф. дис. ... канд. техн. наук / Т. А. Гарифуллин. Уфа : УГАТУ, 2006.
2. **Миронов, В. В.** Концепция динамических XML-документов / В. В. Миронов, Г. Р. Шакирова // Вестник УГАТУ. 2006. Т. 8, № 2(18). С. 58–63.
3. **Миронов, В. В.** Интерпретация XML-документов со встроенной динамической моделью / В. В. Миронов, Г. Р. Шакирова // Вестник УГАТУ. 2007. Т. 9, № 2(20). С. 88–97.

ОБ АВТОРАХ



Миронов Валерий Викторович, проф. каф. автоматиз. систем упр-я. Дипл. радиопизик (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 Рави-левна, 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.