

# Прикладные аспекты в информатике

## Языки XSL и CSS как конструкторы версии для печати в гипертекстовой системе ООСУБД НИКА

А.В. СОЛОВЬЕВ<sup>1,II</sup>, В.А. ТИЩЕНКО<sup>1,II</sup>

<sup>I</sup>Федеральное государственное учреждение «Федеральный исследовательский центр «Информатика и управление» РАН», г. Москва, Россия

<sup>II</sup>Образовательное частное учреждение высшего образования «Православный Свято-Тихоновский гуманитарный университет», г. Москва, Россия

**Аннотация.** Рассматриваются отличия и сходства версии для печати от электронной публикации базы данных на основе ранних разработок, связанных с использованием СУБД НИКА в издательской деятельности. Языки XSL и/или CSS рассматриваются как неотъемлемая часть современных гипертекстовых систем и версии для печати. Делается вывод о том, что генерация фрагментов БД НИКА в виде XML-документов, содержащих объекты форматирования XSL FO или определения стилей CSS для постраничного вывода, позволяет перейти к качественно более высокому уровню отображения объектов БД НИКА в виде PDF-документов.

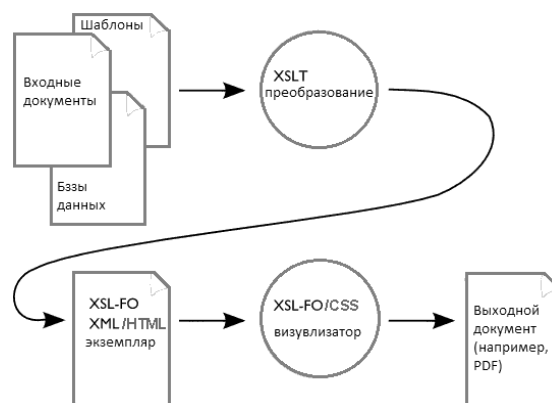
**Ключевые слова:** настольная издательская система, версия для печати, XSLFO/CSS визуализатор, PDF/A документ, CSS Paged Media Module Level 3, спецификации для печати вершин БД НИКА.

**DOI:** 10.14357/20790279200308

### Введение

Версия для печати является важной возможностью электронных изданий баз данных, поскольку создание твердых копий документов – это не только удобная возможность отобразить на бумаге необходимую информацию, но также часто является требованием к подобным системам со стороны пользователя. Версия для печати документов в рамках электронной публикации может быть представлена как в HTML, так и в PDF формате [1]. Формат PDF предпочтителен при сохранении документов, поскольку содержит все составные компоненты, входящие в документ. Приложения могут использовать XSLT-преобразование и объекты форматирования XSL FO или CSS-правила для получения динамически созданного фрагмента БД в PDF-формате (рис.1). Оба

указанных элемента не обязательно присутствуют в цепочке одновременно.



**Рис. 1.** Преобразование входных данных в PDF-документ, используя XML-формат и XSL

Альтернативой объектам форматирования XSL FO служит технология каскадных стилевых таблиц CSS, которая также содержит возможность деления документа на страницы. Эта технология впоследствии призвана заменить объекты форматирования [2]: «CSS-технология становится полной альтернативой по отношению к XSL-FO? – Нет, еще нет, но CSS меняется очень быстро...»

## 1. Система печати баз данных СУБД НИКА

Об отображении БД НИКА в виде документов существует достаточно много работ [3–5]. О видах отображения данных на двумерном листе бумаги говорится в работе [3]. Об издании БД НИКА в виде книжного издания исследуется в работе [4]. В ней рассматривается проблема взаимодействия ООСУБД с электронными издательскими системами. Также отмечается, что модель данных ООСУБД НИКА является более адекватной областям со сложноструктурированными объектами и «нетрадиционными» элементами данных, чем реляционная модель, а издательская деятельность нередко имеет дело именно с такими объектами и данными. Например, макет газеты может содержать типы данных, встречающихся в большей части гипертекстовых сред: текст, графики, пиктограммы, карты и другие типы. Разработанная система печати баз данных СУБД НИКА [4] основана на том, что вершинам схемы описания БД присваиваются спецификации полиграфического набора данных. По тому же принципу строится версия для печати в электронной публикации БД, но только вершинам схемы присваиваются XSL FO объекты или CSS правила. Указанные особенности играют важную роль при построении версии для печати, когда нетерминальная вершина отображается на основе спецификаций в виде последовательности, таблицы, иерархии подчиненных вершин и их комбинаций [3], а к именам вершин и их значениям, входящих в документ, применяются дополнительно XSL FO или CSS стили отображения.

## 2. Отличия и сходства версии для печати и электронной публикации

Спецификация XSL [6] определяет два этапа в представлении данных XML – это преобразование данных для получения результирующего дерева из данных XML и средства форматирования, которые интерпретируют результирующее дерево для отображения данных в соответствующем виде на экране, на бумаге, с помощью речи или другими средствами. Первый этап, связанный с преобразо-

ванием данных, был описан в статье [7] применительно к отображению вершин БД НИКА в формате XML, где стилевые таблицы XSL предназначены для тонкой настройки отображения вершин и играют роль надстройки над ядром, которые составляют спецификации отображения, назначаемые вершинам посредством схемы описания данных. Второй этап предполагает существование языка форматирования, который определяется спецификацией XSL FO [8] или спецификацией каскадных стилевых таблиц CSS [9], которые не входят в язык XSL и применяются к XML/HTML-документам. В XSL FO семантика форматирования вводится на основе каталога классов объектов форматирования. Версия для печати должна представлять собой возможность отображения подмножества вершин БД в некоторый документ в соответствии с определенными спецификациями отображения. Эта задача может быть выполнена на основе применения языка XSL и его реализаций не только в обозревателях интернет, где конечное дерево использует элементы языка HTML и/или каскадные таблицы стилей CSS, но и в виде процессоров объектов форматирования, использующих объекты форматирования XSL FO для преобразования в документ в формате PDF [10]. Подобные процессоры существуют и для отображения на основе CSS-правил, такие как Prince [11] и AN Formatter [12]. Как указано в статье [1] PDF версия имеет следующие преимущества в сравнении с HTML-версией с точки зрения представления данных для печати:

- для создания файла существуют широко распространенные системы набора и верстки;
- возможность распространения и хранения файла;
- является копией печатной версии.

Первый пункт в случае динамического создания PDF предполагает использование соответствующих библиотек или процессоров, генерирующих документ PDF из XML с использованием стилевых таблиц XSL. Во 2-ом подразумевается распространение по электронной почте или на электронном носителе и хранение без необходимости постоянного доступа к ресурсу. 3-й означает, что документ PDF имеет полное сходство с «оригинальной» печатной версией при просмотре на любых устройствах и при печати. В качестве отличий версии для печати от электронной можно взять следующие:

- нет необходимости добавлять гиперссылки;
- нет необходимости добавлять мультимедиа материалы;
- нет необходимости иметь возможность экспорта элементов документа в другие электронные форматы;

- множество других возможностей, связанных, например, с интерактивным взаимодействием пользователя с системой, обслуживающей электронные публикации, также не требуется включать в версию для печати.

При генерации PDF также важно встраивать в документ все используемые в нем шрифты. Для этой цели подходит формат PDF/A. Исходя из преимуществ и отличий PDF формата, используемого в качестве версии для печати, можно сделать вывод [1], что электронная HTML-версия удобна для работы в сети интернет и для быстрого просмотра информации, в то время, как для подробного изучения удобно сохранить данные в PDF-формате и распечатать в виде твердой копии.

Основным сходством между версией для печати и электронной версией является динамически создаваемый документ XML с вершинами БД. Он является одинаковым в обоих случаях и данные идентичны. Другая общая черта связана с использованием языка XSL.

### 3. Средства языка XSL/CSS для формирования версии для печати

Стюарт [13] выделяет ряд преимуществ применения XSL FO:

1. Разделение ответственности. Слой представления данных отделяется от слоя модели.
2. Слабая связь между данными и представлением. Таблицы стилей XSL FO обрабатывают исключительно xml-представление объектов. Подобное разделение между данными и представлением позволяет легко изменять формат документов PDF без изменения объектов БД.
3. Гибкость представления. Подобно CSS XSL FO включает элементы форматирования для управления различными параметрами, связанными с типом, размером шрифта, отступами, оформлением текста.
4. Дополнительные возможности: наборы атрибутов XSLT, поддержка сложного разбиения на страницы, сноски к документу, определение колонтитулов страниц, оптимизация верстки, наследование стилей, оглавление и закладки, поддержка таблиц, поддержка различных направлений записи текста.

Особенностью форматирования с помощью XSL FO объектов или CSS правил является возможность форматирования с точностью до пикселя (в отличие от языка HTML, который без использования CSS может форматировать только с точностью до символа). Объекты форматирования представляют собой часть языка XSL, предназначенные для

отображения XML-документа в печатном виде или в виде электронного документа в формате PDF.

На рис.2 показан пример [14] вложенности формирующих объектов XSL

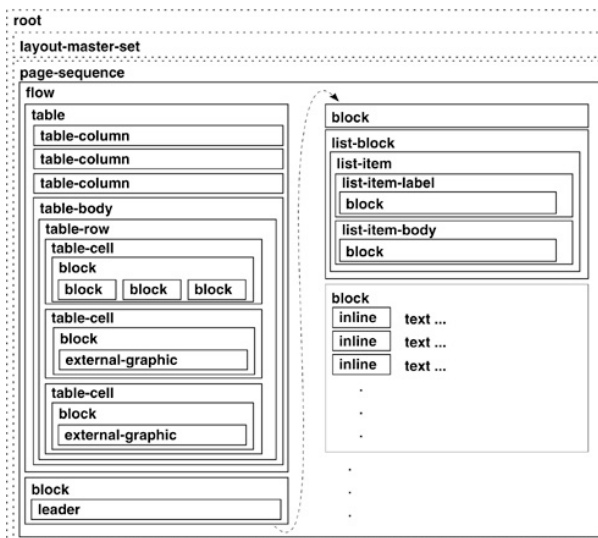


Рис.2. Пример дерева элементов XSL FO

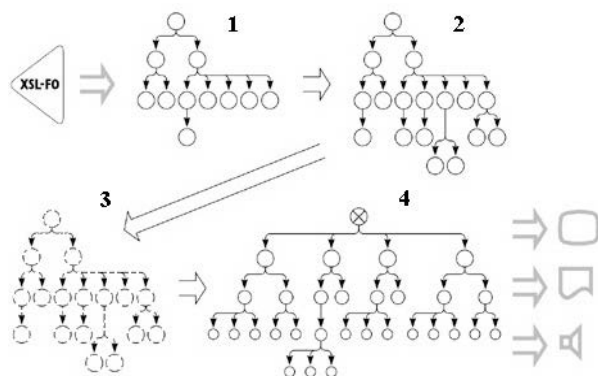


Рис.3. Диаграмма модели обработки XSL

На рис.2 *root* (корневой элемент) – элемент документа в языке XSL FO, который возвращает область, созданные подчиненными элементами, как последовательность областей окна просмотра страницы. *Layout-master-set* (схема набора шаблонов) определяет набор шаблонов страниц, последовательностей страниц, областей. *Page-sequence* (последовательность страниц) определяет информацию для последовательности страниц с общей статической информацией. *Flow* (поток) – содержимое, «заливаемое» на столько страниц, насколько это требуется для его размещения. Подробное описание всех элементов и их свойств в языке XSL FO содержится в [8]

Холмен [14] описывает модель обработки в виде последовательности формальных шагов,

представленных на рис. 3. Здесь 1 обозначает исходное дерево элементов, составленное в соответствии с рекомендацией [8], 2 – это дерево объектов форматирования, в котором текстовые узлы интерпретируются в виде символьных объектов форматирования, а атрибуты узлов – в виде их свойств. При этом из дерева 2 удаляются не имеющие значения для форматирования «пробельные» текстовые узлы, а также объекты, которые не входят в пространство имен XSL FO, например, фрагменты графики SVG. 3 – «утонченное» (переработанное) дерево объектов форматирования, состоящее из объектов и признаков. На основании значения свойств вычисляются значения признаков, например, значение свойства *2em* при текущем размере шрифта *20pt* дает значение признака *40pt*. При этом наследование играет важную роль для вывода признаков. Некоторые признаки выводятся из ближайшей спецификации наследования соответствующих свойств, другие используют точное значение свойства *inherit*. Признаки, не применимые ни к какому объекту, удаляются и каждый объект содержит только те признаки, которые применимы к нему. 4 – дерево областей, полученное из дерева 3, состоит из областей и признаков, соответствующих семантикам объектов. Большинство объектов создают только одну область, некоторые не создают никакой области. Каждая область имеет геометрическую позицию, позицию z-слоя, содержимое, фон, отступы и границы. Вложенность областей в дерево происходит в соответствии с наследованием областей до самой высокой и самой большой области, являющейся областью страницы. Области страниц подчинены корневому узлу в дереве областей. Области страниц упорядочены по их положению в дереве областей, но геометрически не связаны друг с другом. Обозначение XSL-FO на рис.3, предваряющее дерево 1, соответствует шагу генерации исходного документа, содержащего объекты форматирования, а на последнем шаге, следующим за деревом 4, происходит отображение дерева областей на соответствующем устройстве (экране, принтере, проигрывателе).

Модель области в спецификации [8] определяет только прямоугольные области, которые упорядочены в иерархическом порядке в дереве областей. Дочерние области располагаются внутри базовых. Объект форматирования может добавлять области в несколько веток иерархии, размещая дочерние вне базовой области. Объекты данной ветки дерева областей обычно располагаются рядом друг с другом. Существует много различных типов областей, которые определяют различные способы форматирования содержимого областей. Любая

область – это прямоугольник, который содержит отступы со всех сторон, рамку, отступы между рамкой и содержимым и само содержимое. Два понятия – направления письма и ссылка на ориентацию (задается верхняя граница прямоугольника) – определяют визуальное размещение областей на странице. Эти значения определяют направления, в которых добавляются дочерние области в данную, а также стороны до и после области, начальную и конечную стороны. Большинство незаданных свойств для дочерних областей наследуют значения этих свойств от базовых. Блочное и встроенное содержимое добавляется в структуру областей, как определяется свойствами расположения. Подчиненные объекты типа контейнер используются для создания областей, которые переопределяют свойства, определенные базовыми областями. Области на странице могут перекрываться друг с другом полностью или частично. При наложении областей также задается прозрачность фона. Возникает общая проблема форматирования, состоящая в том, что области из разных ветвей дерева могут занимать одно и то же положение на странице. Необходимо планировать заранее расположение областей, чтобы их расположение в одной ветке не влияло на расположение в других ветках. Страница в XSL-FO определяется ее размерами, различными зонами (5 зон – содержимое документа, перед, после зоны содержимого, начальная и конечная зона) и областями, в которых информация размещается объектами форматирования. Ссылки на зоны задаются с помощью их имен. Все 5 зон на странице могут содержать как статическое содержимое, так и поток информации. Постраничный поток создает столько страниц, сколько необходимо для данного потокового содержимого. Статическое содержимое может иметь динамические компоненты, например, номера страниц, содержимое символов разметки, определенных пользователем. Модель областей является надстройкой над моделью форматирования областей, определенной в спецификации для каскадных стилевых таблиц, включает ссылки на области из других областей, определяет связи и интервалы между областями, сгенерированными буквами, словами, строками и блоками. Области могут быть следующих типов:

- ссылочная (изменяет ориентацию и режим письма);
- окна просмотра (отсеченная/прокручиваемая видимая часть ссылочной области);
- размещения (основа расположения и выравнивания содержимого в пределах базовой области);
- блочная (множество строк, расположенное в заданном направлении);



- линейная (набор встроенных структур в блоке);
- встроенная (атомарная структура или набор глифов в заданном направлении);
- глифа (единственный символ текста).

На рис.4 [8] для примера показаны в виде узлов дерева окна просмотра и ссылочные области для страниц, зон, потоков, сносок и главного потока.

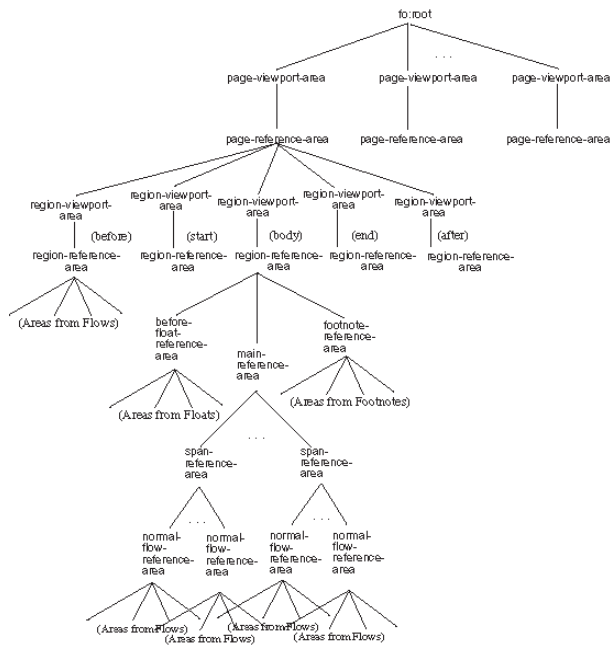


Рис.4. Пример дерева областей

С точки зрения элементов языка XSL основные пять областей или зон страницы определяются с помощью соответствующих элементов, задаваемых шаблоном страницы fo:simple-page-master внутри схемы набора шаблонов fo:layout-master-set:

- fo:region-before – передняя зона (верхний колонтитул);
- fo:region-after – задняя зона (нижний колонтитул);
- fo:region-body – центральная зона (содержимое страницы);
- fo:region-start – начальная зона (левая часть страницы);
- fo:region-end – конечная зона (правая часть страницы).

Кроме шаблонов страниц fo:simple-page-master, элемент fo:layout-master-set может содержать шаблоны последовательностей страниц fo:page-sequence-master, которые могут быть заданы в виде единственного шаблона страницы fo:single-page-master-reference (ссылающегося на fo:simple-page-master), в виде повторения единственного шаблона страницы fo:repeatable-page-master-reference

и в виде повторения множества шаблонов страниц fo:repeatable-page-master-alternatives. Страницы создаются посредством одного или нескольких элементов последовательностей страниц fo:page-sequence, которые как и элемент fo:layout-master-set подчинены корневому элементу fo:root. Каждый элемент fo:page-sequence ссылается на соответствующий шаблон страницы в схеме набора шаблонов. Содержимым страницы управляют объекты потока fo:static-content (статическое содержимое – колонтитулы) и fo:flow (содержимое документа – основной поток информации). К этим элементам применим атрибут flow-name – зона размещения потока. Он принимает следующие значения, соответствующие упомянутым пяти зонам страницы: xsl-region-body, xsl-region-before, xsl-region-after, xsl-region-start, xsl-region-end, а также xsl-before-float-separator, xsl-footnote-separator. Последние два значения атрибута применимы к элементу fo:static-content и определяют разделители, которые располагаются непосредственно перед содержимым плавающей области fo:float и после плавающей области fo:float соответственно. В одной зоне не может быть двух потоков, статическое содержимое fo:static-content предшествует основному fo:flow. Содержимое элемента fo:flow – это набор областей в виде блоков fo:block. Внутри блоков могут располагаться опять блоки fo:block, а также внутрискочные форматирующие объекты, которым соответствуют встроенные области:

- fo:character – символ;
- fo:external-graphic – внешняя графика, встраиваемая в PDF-документ;
- fo:initial-property-set – форматирование первой строки;
- fo:instream-foreign-object – вставка встроенной графики или другого объекта (SVG, MathML);
- fo:inline – встроенная область (часть текста в блоке);
- fo:inline-container – группирование встроенных элементов для совместной обработки;
- fo:leader – пунктиры и линейки;
- fo:page-number – номер страницы.

Таким образом, структуру документа схематично можно описать следующим образом:

| Документ | = | Набор страниц   |
|----------|---|---|
| Страница | = | набор зон (заголовки, основная часть, конец)                      |
| Зона     | = | набор областей (блок, таблица, список)                            |
| Область  | = | набор областей   строка   |
| Строка   | = | набор внутри-строчных элементов (символы, сноски, мат. выражения) |

Кроме перечисленных объектов существуют также внестрочные формирующие объекты:

- fo:float – плавающий блок, отдельная, незакрепленная область в начале страницы или сдвинутая к одной из сторон;
- fo:footnote – ссылка на сноску и связанную с ней сноску;
- fo:footnote-body – текст сноски.

Наконец, гипертекстовые ссылки создаются с помощью элемента fo:basic-link.

Для сравнения спецификация CSS [9] также определяет модель страницы для вывода документа на печать или преобразование в PDF-документ. В модели форматирования постраничного носителя документ переносится в одну или несколько прямоугольных областей страниц. Свойства страниц и областей могут быть заданы с помощью ат-правил, представляющих собой команды или инструкции для выполнения или поведения [15]. Спецификацией CSS определяется порядок применения каскада правил к элементам посредством приоритетов и специфичности. В CSS делается различие между левой и правой страницами, являются ли они двухсторонними или нет, также могут быть выделены первая страница, пустые страницы, поименованные страницы для определения типа страницы. CSS определяет свойства для указания в каких местах должны быть или могут быть разрывы страниц. В этой модели страница кроме области содержимого имеет 16 зон, представляющих собой колонтитулы страницы, расположенные вокруг содержимого страницы и прорисовываемые в последнюю очередь по умолчанию в порядке увеличения номеров по часовой стрелке (рис. 5):

- 1 – @top-left-corner,
- 2 – @top-left,
- 3 – @top-center,

- 4 – @top-right,
- 5 – @top-right-corner,
- 6 – @right-top,
- 7 – @right-middle,
- 8 – @right-bottom,
- 9 – @bottom-right-corner,
- 10 – @bottom-right,
- 11 – @bottom-center,
- 12 – @bottom-left,
- 13 – @bottom-left-corner,
- 14 – @left-bottom,
- 15 – @left-middle,
- 16 – @left-top.

#### 4. Отображение данных БД НИКА в виде версии для печати

После краткого изложения модели работы XSL и его элементов, необходимо применить этот язык для отображения вершин БД НИКА в виде PDF-документов. Ниже приводятся примеры шаблонов XSL-FO такого применения.

Перед рассмотрением примеров необходимо отметить, что существенным преимуществом использования технологии XSL FO и XML является отделение представления данных от самих данных. Это неоднократно подчеркивал проф. Н.Е. Емельянов в отношении языка OOML [16]. Этот фундаментальный метод позволяет менять представления данных, не внося изменений в сами объекты БД НИКА. Кроме того, одно и то же представление можно применять для отображения разных объектов. С этой точки зрения XSL FO в сочетании с XSLT [7] – это некоторая возможная схема XML для отображения данных в формате PDF. Эта схема состоит из шаблонов преобразования элементов XML в объекты форматирования XSL FO. Язык XPath, который используется в XSLT, позволяет применять шаблоны в разном контексте и с определенными атрибутами. Об отображении объектов БД НИКА в виде XML-документов говорится в статье [7]. Таких способов отображения может быть множество. Одним из них является естественный способ представления в виде XML-документа, в котором имена элементов – это названия объектов и их свойств в БД НИКА, а содержимое элементов – это иерархия подчиненных объектов в виде вложенных XML-элементов и значения свойств [17]. Таким образом, схема представления объектов БД в виде объектов форматирования будет представлять собой набор шаблонов с образцами выбора (атрибут match), совпадающими с именами объектов БД НИКА. Кроме того, все особенности работы с XSL, описанные в статье Стюарта [13] (см. раздел «Средства языка XSL/CSS для формирования версии для

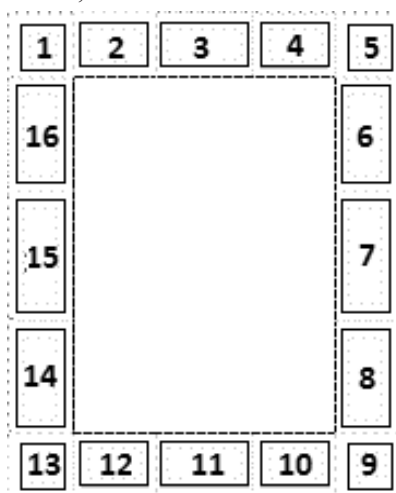


Рис. 5. Порядок прорисовки колонтитулов, определяемых спецификацией CSS

печати)), могут быть учтены при составлении XSL шаблонов для вершин БД НИКА. При этом для создания разных шаблонов отображения для одного и того же объекта может быть использован атрибут `spes`, в котором указываются спецификации отображения для данного объекта, определенные в схеме БД НИКА. Также Стюарт дает рекомендации по использованию XSLT и XSL FO [13].

- Необходимо преимущественно применять шаблоны с образцами выбора с атрибутом `match`, а не именованные шаблоны с атрибутом `name`, т.к. преобразование наборов вершин лучше соответствует декларативной природе XSLT, чем вызов процедур.
- По той же причине применение рекурсии также является более предпочтительным во избежание излишнего усложнения шаблонов.
- Предварительно необходимо детально разработать схему XML.

При отображении объектов БД НИКА последний пункт можно считать выполненным, т.к. в данном случае схема XML совпадает со схемой БД. Также важно при отображении объектов БД не включать в XML-документ те объекты, которые не участвуют в отображении посредством спецификации ОМТ (не отображать данный объект) [5], что позволяет уменьшить размер XML-документа, передаваемого по сети.

Во фрагменте шаблона `dcmt` (рис.6) для отображения биографических справок фактографической БД по репрессированным [18] в виде двух колонок текста на странице PDF-документа используется атрибут `column-count="2"` в элементе `fo:region-body` при задании шаблона страницы.

Пример результата применения CSS и HTML в результате XSLT-преобразования приводится в виде окна обозревателя на рис.7.

В примере отображение по колонкам задается посредством CSS-свойств `column-count:2` и `column-gap:1rem`.

## 5. Рекомендации при построении версии для печати

Д. Стори формулирует в своей статье [19] положения, руководствуясь которыми можно задать версии для печати:

1. Посредством медиа-запроса `@media print` выделяются отличия версии для печати:
  - точные определения цветов;
  - пропуск интерактивных элементов на странице посредством `display:none`;
  - использование единиц измерения сантиметров и дюймов;

```

<xsl:template match="//dcmt">
  <fo:root>
    <fo:layout-master-set>
      <fo:simple-page-master master-name="pdf-document" page-height="29.7cm" page-width="21cm" margin-top="1cm" margin-bottom="2cm" margin-left="2.5cm" margin-right="2.5cm">
        <fo:region-body margin-top="3cm" column-count="2" column-gap="0.6325cm"/>
        <fo:region-before extent="3cm"/>
        <fo:region-after extent="1.5cm"/>
      </fo:simple-page-master>
    </fo:layout-master-set>
    <fo:page-sequence master-reference="pdf-document">
      <fo:static-content flow-name="xsl-region-before">
        <fo:block line-height="14pt" font-size="10pt" color="gray" font-family="Calibri" text-align="end">Биографические справки</fo:block>
      </fo:static-content>
      <fo:static-content flow-name="xsl-region-after">
        <fo:block line-height="14pt" font-size="10pt" color="gray" text-align="end">
          Page
          <fo:page-number/>
        </fo:block>
      </fo:static-content>
      <xsl:apply-templates select="//dcmt/cnt[1]"/>
    </fo:page-sequence>
  </fo:root>
</xsl:template>

```

Рис.6. Пример шаблона `dcmt` для отображения объектов БД НИКА в две колонки

- использование всей поверхности страницы посредством `width:100%`, `padding:0`, `margin:0`;
  - определение полей страницы при помощи ат-правила `@page {margin: 1.5cm}`;
  - регулировка переносов страниц с помощью `page-brak-before`, `page-break-after` и `page-break-inside` так, чтобы избежать обрыв страниц на заголовках, изображениях, списках, разделы начинались с новой страницы;
  - масштабирование изображения при необходимости – `max-width:100%`.
2. Установка фона изображений и цветов.
  3. Печать внешних гипертекстовых ссылок: `article a[href^=http]:after {content:"<" attr(href) ">"};` и пропуск ссылок изображений `$a:after > img {content: " "};`. Последнее правило позволяет задать CSS уровня 4, также внешние ссылки могут быть легко заданы – `a:not(:local-link):after {content: "<" attr(href) ">"};`.
  4. Использование QR-кодов для автоматического ввода ссылок. Для этого можно использовать API генераторов QR-кодов, которые на основе текста ссылки формируют изображение QR-кода. Например, ссылка на биографическую справку иеромонаха Анастасия (Колтаков Александр Семенович) `:http://martyrs.pstbi.ru/bin/db.exe/no_dbpath/docum/cnt/css_nmnden/ans/nm1/?HYZ9EJxGHoxITYZCF2JMTWhyPn*n9X2mCE*` дает следующий QR-код:



**КОЧНЕВА Прасковья Степановна (Максимовна), член церковной двадцатки**



1890 — родилась в населённом пункте: Оренбургская губ. с.Кочнево.

По данным [1,2] отчество Максимовна, родилась в с.Левжа Рузаевского района Мордовии. По данным [3]: Из семьи крестьянина Степана Кочнева. Окончила сельскую школу и жила с родителями

1932 — Служение. Оренбургская губ., с.Кочнево, церковь  
Прислуживала в храме в родном селе. В 1932г. власти потребовали от Параскевы, ревностно защищавшей храм от закрытия, чтобы она покинула село. И она переехала в г.Миасс Челябинской области, где устроилась сторожем при храме св. князя Александра Невского

1932 — Служение. Челябинская о., г.Миасс, храм св. благоверного князя Александра Невского

Из-за размолвки с псаломщицей, нарушавшей, по мнению Параскевы, церковный устав, она уехала в село Вознесенское, где стала помогать в храме

Служение. Челябинская о., с.Вознесенское, храм Вознесения Господня

Через год Параскева вернулась в Миасс, в Александро-Невский храм, и ее выбрали членом церковной двадцатки

1937 — Служение. Челябинская о., г.Миасс, храм св. благоверного князя Александра Невского

В 1937г. был арестован священник Александро-Невского храма, и верующие стали искать другого священника, но всех, кого ни находили, отказывались регистрировать местные власти, и Параскева предложила членам двадцатки написать жалобу во ВЦИК. Приехав в Москву 27 декабря 1937г., она подала жалобу чиновникам ВЦИКа, а сама отправилась на прием к заместителю Местоблестителя митрополиту Сергию (Страгородскому) с просьбой прислать им священника. Митрополит Сергей благословил ехать служить в храм Александра Невского сидевшего в приемной и просившего места священника. Из Патриархии они сразу же отправились на Казанский вокзал, где купили билеты, собираясь через несколько дней выехать в Миасс

1937, 29 декабря — арестована. Москва, Казанский вокзал.  
Священник этот был родом из Бессарабии, отошедшей в то время к Румынии, и он имел намерение переехать из советской России на родину. В один из приездов в Москву он подал документы в Румынское посольство, и теперь,

решив использовать время до отъезда, отправился в консульство, чтобы узнать, нет ли ответа на его прошение о въезде в Румынию. Это было 29 декабря. После посещения консульства священник отправился на Казанский вокзал — здесь он был арестован, а вместе с ним Параскева. Они были заключены в Бутырскую тюрьму и в тот же день допрошены

1938, 14 февраля — осуждена. Тройка при НКВД за "а/с деятельность" (по ст.58-10,58-11 УК РСФСР). Приговор — 10 лет ИТЛ.

Из протокола допроса 29 декабря 1937г.: Вопрос: "Вы являетесь членом двадцатки церкви Александра Невского?" Ответ: "Я являюсь активной церковницей, организовавшей подписи под заявлением, которое мною было передано во ВЦИК, и являюсь членом двадцатки храма Александра Невского". Вопрос: "Следствие располагает данными, что Вы среди окружающих вас лиц вели к/р агитацию и распространяли провокационные слухи о скорой гибели советской власти и втором пришествии Христа. Вы это подтверждаете?" Ответ: "Нет, я это отрицаю". Вопрос: "Уточните, чем Вы занимались до ареста?" Ответ: "До ареста я занималась исключительно охраной церкви — сторожила. Кроме того, я состояла в церковной двадцатке"

1937, 29 декабря — Отбывала заключение. Москва, Бутырская тюрьма

1939, 8 апреля — Отбывала заключение. Амурская о., Буреинский р., ст.Известковая, Бамлаг, Буреинский желдорлаг  
Параскева Кочнева была раздавлена во время работы и скончалась в лагерьном лазарете от ран 8 апреля 1939г.

1939, 8 апреля — дата кончины. Причина смерти — погибла в ИТЛ. Место смерти — Амурская о., Буреинский р., ст.Известковая, Бамлаг, лазарет.

*\*\* Выписка из журнала заседания Священного Синода Русской Православной Церкви от 16 июля 2005г.*

*База данных о жертвах политического террора в СССР. Компакт-диск. 3-е изд. НИИЦ "Мемориал". М.:Звенья, 2004.627078*

*Память: жертвы политических репрессий. Саранск. Мордовское книжное издательство. 2000.*

*http://www.eparh.ru/content/view/113/63/ (сайт Биробиджанской епархии).*

*http://www.butyrka.st-nikolas.ru/ (сайт Покровского храма в Бутырской тюрьме).*

**Рис. 7.** Пример результата отображения биографической справки из БД посредством XSLT, HTML и CSS.

5. Использование CSS-фильтров для улучшения качества печати посредством инвертирования изображений, например, `img {filter:invert(100%);}`.

В заключении необходимо отметить, что CSS в сравнении с XSL FO имеет на данный момент меньшие возможности для создания версии для печати, например, в CSS отсутствуют схемы набора шаблонов, которые позволяют организовывать более сложную верстку документов. Основной вывод состоит в том, что обе технологии XSL FO и CSS позволяют динамически формировать PDF-документы, содержащие фрагменты БД НИКА.

**Литература**

1. Чебуков Д.Е. Об HTML версии полного текста научной статьи // Научный сервис в сети Интернет: труды XX Всероссийской научной конференции (17-22 сентября 2018 г., г. Новороссийск). – М.: ИПМ им. М.В.Келдыша, 2018. С. 487-498. <http://keldysh.ru/abrau/2018/theses/16.pdf> (Доступ 20.07.2020).

2. *Quin L* (November 2, 2013). Re: [xsl] xsl 2.0? <http://www.biglist.com/lists/lists.mulberrytech.com/xsl-list/archives/201311/msg00012.html> (Доступ 20.07.2020).

3. *Емельянов Н.Е.* Теоретический анализ документного интерфейса / Н.Е. Емельянов. М.: ВНИИСИ. 1987. 40 с.

4. *Емельянов Н.Е., Соловьёв А.В.* Использование СУБД в издательской деятельности // Материалы VI Международной конференции «Применение новых технологий в образовании» 29 июня – 2 июля 1995 г. Троицк. С. 123-125.

5. *Емельянов Н.Е., Тищенко В.А.* Методы отображения объектов для построения web-сервера объектно-ориентированной базы данных. Развитие безбумажных технологий в организационных системах // Труды ИСА РАН. М.: URSS. 1999. С. 96-109.

6. XSL Transformations (XSLT) Version 3.0. W3C Recommendation 8 June 2017. <https://www.w3.org/TR/xslt-30/> (Доступ 20.07.2020).

7. *Тищенко В.А.* Применение языка XSL для отображения БД НИКА / В.А. Тищенко // Организационное управление и искусственный интеллект. Труды ИСА РАН. М.: URSS. 2003. С. 149-175.



8. Extensible Stylesheet Language (XSL) Version 1.1. W3C Recommendation 05 December 2006. <https://www.w3.org/TR/xsl11/> (Доступ 20.07.2020).
9. CSS Paged Media Module Level 3. W3C Working Draft, 18 October 2018. <https://www.w3.org/TR/css-page-3/> (Доступ 20.07.2020).
10. *Giannetti F., Fernandes L.G., Timmers R., Nunes T., Raeder M., Castro M.B.* High performance XSL-FO rendering for variable data printing. SAC 2006: Proceedings of the 2006 ACM symposium on Applied computing. P. 811-817.
11. *Day M.* Printing XML: Why CSS is better than XSL, May 2005, a paper presented at the XTech 2005 Conference in Amsterdam.. XTech 2005, Gilbane Conference to Co-locate in Amsterdam 24-27 May.
12. Introduction to CSS for Paged Media. <https://www.antennahouse.com/css> (Доступ 20.07.2020).
13. *Стюарт Б.* Создание документов в формате PDF при помощи XStream и XSL-FO. 07.12.2011. <https://www.ibm.com/developerworks/ru/library/x-xstrmfo/index.html> (Доступ 20.07.2020).
14. *Holman G. K.* Definitive XSL-FO, Prentice Hall PTR, March 21, 2003. P. 480.
15. *Graham G.* The At-Rules of CSS. Last Updated May 13, 2015. <https://css-tricks.com/the-at-rules-of-css/> (Доступ 20.07.2020).
16. *Bogacheva A.N., Emeljanov N.E., Romanov A.P.* Object Oriented Markup Language and Restructuring Hierarchical Database Objects // Proceeding ADBIS '95 Proceedings of the Second International Workshop on Advances in Databases and Information Systems. pp. 137-142, June 27 - 30, 1995.
17. *Богданов А.С., Емельянов Н.Е., Ерохин В.И., Романов Б.Л.* Реализация запросной системы на основе XPath для ООСУБД НИКА // Труды ИСА РАН. М.: Едиториал УРСС. 2003.С. 130-146.
18. База данных «За Христа пострадавшие». <http://martyrs.pstbi.ru/> (Доступ 20.07.2020).
19. *Storey D.* 5 Powerful Tips And Tricks For Print Style Sheets, Smashing Magazine, march 8, 2013 <https://www.smashingmagazine.com/2013/03/tips-and-tricks-for-print-style-sheets/> (Доступ 20.07.2020).

**Соловьев Александр Владимирович.** Федеральное государственное учреждение «Федеральный исследовательский центр «Информатика и управление» РАН», Москва, Россия. Заведующий лабораторией. Доктор технических наук. Количество печатных работ: более 32. Область научных интересов: фундаментальные проблемы организации электронного документооборота. E-mail: [soloviev@isa.ru](mailto:soloviev@isa.ru) (Ответственный за переписку)

**Тищенко Владимир Александрович.** Федеральное государственное учреждение «Федеральный исследовательский центр «Информатика и управление» РАН», Москва, Россия. Научный сотрудник. Количество печатных работ: 20. Область научных интересов: средства создания и поддержки электронных библиотек и электронных изданий. E-mail: [vtischenko@isa.ru](mailto:vtischenko@isa.ru)

## XSL and CSS as printable version constructors in a hypertext system for NIKA DBMS

Soloviev A.V.<sup>1,II</sup>, Tishchenko V.A.<sup>1,II</sup>

<sup>1</sup> Federal Research Center "Informatics and control" of The Russian Academy of Sciences, Moscow, Russia

<sup>II</sup> St. Tikhons' Orthodox University, Moscow, Russia

**Abstract.** Differences and similarities between the print version and the electronic publication of a database based on early developments related to the use of the NIKA DBMS in publishing are considered. XSL and / or CSS are considered to be an integral part of modern hypertext systems and print versions. It is concluded that the generation of NIKA database fragments in the form of XML documents containing XSL FO formatting objects or CSS style definitions for paginated output allows you to move to a qualitatively higher level of displaying NIKA database objects as PDF documents.

**Keywords:** *desktop publishing system, print version, XSLFO/CSS renderer, PDF/A document, CSS Paged Media Module Level 3, specifications for printing vertices of NIKA DB.*

**DOI:** 10.14357/20790279200308

## References

1. *Chebukov D.E.* On the HTML version of the full text of the scientific article // Scientific service on the Internet: proceedings of the XX All-Russian Scientific Conference (September 17-22, 2018, Novorossiysk). - М.: IPM im. M.V. Keldysh, 2018. -- doi: 10.20948 / abrau-2018-16 - p. 487-498. <http://keldysh.ru/abrau/2018/theses/16.pdf> (Accessed 20.07.2020).
2. *Quin L* (November 2, 2013). Re: [xsl] xsl 2.0? <http://www.biglist.com/lists/lists.mulberrytech.com/xsl-list/archives/201311/msg00012.html> (Accessed 07/20/2020).
3. *Emelyanov N.E.* Theoretical analysis of the document interface / N.E. Emelyanov. M., preprint VNIISI, 1987, 40 p.
4. *Emelyanov N.E., Soloviev A.V.* The use of DBMS in publishing // Proceedings of the VI International conference «The use of new technologies in education» June 29 - July 2, 1995 Troitsk. - S. 123-125.
5. *Emelyanov N.E., Tishchenko V.A.* Object mapping methods for building an object-oriented database web server. Development of paperless technologies in organizational systems // Proceedings of the ISA RAS. Ed. Doctor of Technical Sciences prof. V.L. Arlazarov and Doctor of Technical Sciences prof. NOT. Emelyanova - М.: URSS. 1999. S. 96-109.
6. XSL Transformations (XSLT) Version 3.0. W3C Recommendation 8 June 2017. <https://www.w3.org/TR/xslt-30/> (accessed 07/20/2020).
7. *Tishchenko V.A.* The use of XSL for displaying the NIKA database / V.A. Tishchenko // Organizational management and artificial intelligence. Proceedings of the ISA RAS. Ed. Doctor of Technical Sciences prof. Arlazarova V.L. and d.t.s. prof. Emelyanova N.E. M.: URSS. 2003. S. 149-175.
8. Extensible Stylesheet Language (XSL) Version 1.1. W3C Recommendation 05 December 2006. <https://www.w3.org/TR/xsl11/> (Accessed 20.07.2020).
9. CSS Paged Media Module Level 3. W3C Working Draft, 18 October 2018. <https://www.w3.org/TR/css-page-3/> (Accessed 20.07.2020).
10. *Giannetti F., Fernandes L.G., Timmers R., Nunes T., Raeder M., Castro M.B.* High performance XSL-FO rendering for variable data printing. SAC 2006: Proceedings of the 2006 ACM symposium on Applied computing. P. 811-817
11. *Day M.* Printing XML: Why CSS is better than XSL, May 2005, a paper presented at the XTech 2005 Conference in Amsterdam .. XTech 2005, Gilbane Conference to Co-locate in Amsterdam 24-27 May
12. Introduction to CSS for Paged Media. <https://www.antennahouse.com/css> (Accessed 07/20/2020).
13. *Stewart B.* Create PDF documents using XStream and XSL-FO. 07.12.2011. <https://www.ibm.com/developerworks/ru/library/x-xstrmf/index.html> (Accessed 07/20/2020).
14. *Holman G. K.* Definitive XSL-FO, Prentice Hall PTR, March 21, 2003, ISBN: 0-13-140374-5, P. 480.
15. *Graham G.* The At-Rules of CSS. Last Updated May 13, 2015. <https://css-tricks.com/the-at-rules-of-css/> (Accessed 07/20/2020).
16. *Bogacheva A.N., Emeljanov N.E., Romanov A.P.* Object Oriented Markup Language and Restructuring Hierarchical Database Objects // Proceeding ADBIS '95 Proceedings of the Second International Workshop on Advances in Databases and Information Systems. pp. 137-142, June 27-30, 1995.
17. *Bogdanov A.S., Emelyanov N.E., Erokhin V.I., Romanov B.L.* Implementation of a query system based on XPath for OODBMS NIKA // Sb. Proceedings of the ISA RAS. Ed. Corresponding Member RAS Arlazarov V.L. and d.t.s. prof. Emelyanova N.E. - М.: Editorial URSS, 2003. - S. 130-146.
18. Database "Suffered for Christ". <http://martyrs.pstbi.ru/> (Accessed 20.07.2020).
19. *Storey D.* 5 Powerful Tips And Tricks For Print Style Sheets, Smashing Magazine, March 8, 2013 <https://www.smashingmagazine.com/2013/03/tips-and-tricks-for-print-style-sheets/> (Access 20.07.2020).

**A.V. Soloviev.** d.t.s., Main researcher, FRC CSC RAS. Head of lab. IRS Faculty of IAM, PSTGU. Graduated from the MSTU N.E. Bauman in 1993. The number of printed works is more than 32. Research interests: fundamental problems of organization of electronic document management. E-mail: [soloviev@isa.ru](mailto:soloviev@isa.ru) (Responsible for the correspondence)

**V.A. Tishchenko.** Researcher, RC CSC RAS. Employee of lab. IRS Faculty of IAM, PSTGU. Graduated from the MEPhI in 1993. Number of publications: 17. Research interests: means of creation and support of electronic libraries and electronic publications. E-mail: [vtishchenko@isa.ru](mailto:vtishchenko@isa.ru)