

Представление гипертекста в СУБД НИКА

Н. Е. Емельянов, В. А. Тищенко

*Институт системного анализа Российской академии наук,
Россия, 117312 Москва, пр. 60-летия Октября, 9*

Описываются спецификации данных для представления документов средствами СУБД НИКА. Вводятся дополнительные методы отображения html-документа посредством html/xml-сервера СУБД НИКА. Приводится пример использования xsl-шаблона при отображении вершин базы данных в виде xml-документов. Модель системы представляется в виде трехуровневой гипертекстовой машины: пользовательский интерфейс, гипертекстовая машина, база данных. Рассматриваются методы обхода сложных структур данных, применяемые к гипертексту: прокрутка, просмотр, поиск, исследование, случайные передвижения. Рассматривается двойственность структуры базы данных и структуры html/xml-документов, а также построение динамической структуры гипертекста в виде сетей Петри.

Гипертекст позволяет работать с текстом не только в последовательном порядке, как с обычным текстом, но и в порядке, определяемом топологией гипертекста, т. е. переходя по выделенному понятию на определенное место в тексте, связанное по смыслу с данным понятием. Для применения инструментальных средств и возможностей СУБД к работе с гипертекстом является важным построение гипертекста на основе СУБД.

1. Ядро гипертекстовой системы на основе СУБД НИКА

Одним из приложений СУБД НИКА является гипертекстовая система, построенная на ее основе. Вызвано это не только широким распространением гипертекста, но также и близостью модели данных СУБД НИКА (сеть с выделенными иерархиями вершин) к модели данных гипертекстовой системы [1]. В статье о построении html/xml-сервера на основе базы данных НИКА дается описание примера такой гипертекстовой системы [2], где данные из БД НИКА отображаются в виде гипертекста по-

средством спецификаций отображения вершин, присваиваемых вершинам схемы базы данных. Однако есть существенное отличие гипертекстовой модели данных, построенной на языке HTML, от модели данных СУБД НИКА. Гипертекстовая система в Интернете использует слабоструктурированную модель данных, когда данные описывают «сами себя» в том смысле, что они не имеют отдельной схемы, и структура данных, если она существует, вытекает из самих данных. Такая модель данных может быть полезна, когда интегрируется несколько баз данных, часть из которых не структурирована [3]. В СУБД НИКА [4] используется полностью структурированный подход к данным. Иерархия вершин-объектов описывается в виде отдельной схемы данных. В [3] в качестве слабоструктурированной модели данных предлагается гипервершинная модель базы данных. Гипервершины — это графы, вершины которых могут содержать графы, кроме обычных вершин. В данной статье будет рассмотрен метод представления гипертекста посредством СУБД НИКА, построенной на основе типовой полной сложноструктурированной модели данных.

Гипертекст, построенный с помощью языка HTML (XML), представляет собой граф, вершины которого — html/xml-документы. Каждый html-документ состоит из иерархии html-элементов. Такой гипертекст может быть отображен в базу данных НИКА с помощью соответствующей схемы данных. Эта схема может быть определена следующим образом в соответствии с определением типа базы данных в статье [5]. Пусть множество вершин V есть объединение непересекающихся подмножеств терминальных вершин V^* , вершин массивов $V^\#$, вершин структур $V^@$, вершин ссылок на шаблон V^p и вершин ссылок на значение V^r [5]. Вершина v_1 представляет собой вершину верхнего уровня БД — массив документов, т. е. $v_1 \in V^\#$, $(V_0, v_1) \in E(V)$, где V_0 — корневая вершина и $E(V)$ — множество дуг графа, соответствующего типу объекта БД (рис. 1). Элемент массива el_{v_1} имеет множество подчиненных элементов $E(el_{v_1})$, представляющего из себя структуру типа $T(el_{v_1}) = \{T(key_{v_1}), T(v_2), T(v_3), T(v_4)\}$, состоящую из последовательности типов подчиненных вершин, где key_{v_1} : $key_{v_1} \in V^*$ — ключ элемента массива v_1 , являющегося номером или идентификатором документа; v_2 : $v_2 \in V^\#$ — массив html-элементов; v_3 : $v_3 \in V^*$ — значение ключа корневого html-элемента или элемента с меткой html в массиве v_2 ; v_4 : $v_4 \in V^r$, $v_4 = ref_{v_2}(v_3)$ — ссылочная вершина на значение, осуществляющая переход на корневой html-элемент в массиве v_2 по ключу v_3 . Элемент массива el_{v_2} имеет множество подчиненных элементов $E(el_{v_2})$ и описывается типом

$$T(el_{v_2}) = \{T(key_{v_2}), T(v_5), T(v_6), T(v_7), T(v_8), T(v_9), T(v_{10}), T(v_{11})\},$$

где key_{v_2} : $key_{v_2} \in V^*$ — ключ элемента массива v_2 , являющегося идентификатором html-элемента (уникальным для данного документа); v_5 : $v_5 \in V^@$ — структура, содержащая значения метки и атрибутов для данного html-эле-

мента; $v_6: v_6 \in V^*$ — текстовое содержимое для элементов, содержащих текст; $v_7: v_7 \in V^\#$ — массив строк — продолжение текста, определяемого вершиной v_6 (свыше 255 символов); $v_8: v_8 \in V^*$ — идентификатор первого подчиненного html-элемента данному элементу в массиве html-элементов v_2 ; $v_9: v_9 \in V^I$, $v_9 = \text{ref}_{v_2}(v_8)$ — ссылочная вершина на значение, осуществляющая переход на заданный html-элемент в массиве v_2 по ключу v_8 ; $v_{10}: v_{10} \in V^*$ — идентификатор следующего html-элемента на данном уровне иерархии в массиве html-элементов v_2 ; $v_{11}: v_{11} \in V^I$, $v_{11} = \text{ref}_{v_2}(v_{10})$ — ссылочная вершина на значение, осуществляющая переход на заданный html-элемент в массиве v_2 по ключу v_{10} . Окончательно получаем следующую структуру:

$$\begin{aligned}
 T(V_0) = \{ & [T(\text{key}_{v_1}), \\
 & \{[T(\text{key}_{v_2}), \\
 & \quad \{T(v_{12}), \{[T(\text{key}_{v_{13}}), T(v_{14}), T(v_{15})]\}\}, \\
 & \quad T(v_6), \\
 & \quad \{[T(\text{key}_{v_7}), T(v_{17})]\}\}, \\
 & \quad T(v_8), T(\text{ref}_{v_2}(v_8)), T(v_{10}), T(\text{ref}_{v_2}(v_{10}))\}], \\
 & T(v_3), \\
 & T(\text{ref}_{v_2}(v_3))\}], \quad (1)
 \end{aligned}$$

где квадратные скобки обозначают элемент массива, т. е. $T(v_i) = [T(\text{el}_{v_i})]$, где $v_i \in V^\#$. Здесь $T(v_5) = \{T(v_{12}), \{[T(\text{key}_{v_{13}}), T(v_{14}), T(v_{15})]\}\}$, где $v_{12}: v_{12} \in V^*$ — метка данного html-элемента; $\text{key}_{v_{13}}: \text{key}_{v_{13}} \in V^*$ — ключ элемента нумерованного массива v_{13} атрибутов и значений атрибутов html-элемента; $v_{14}: v_{14} \in V^*$ — атрибут html-элемента; $v_{15}: v_{15} \in V^*$ — значение атрибута html-элемента; $T(v_7) = \{[T(\text{key}_{v_7}), T(v_{17})]\}$, где $\text{key}_{v_7}: \text{key}_{v_7} \in V^*$ — ключ элемента нумерованного массива строк v_7 ; v_{17} — текстовая строка массива. Для реализации гипертекстовых ссылок средствами СУБД НИКА (а не только с помощью гипертекстового элемента «а» с заданным атрибутом «href» со значением ссылки в виде URL) полученную структуру (1) необходимо в элементе массива html-элементов el_{v_2} дополнить ссылкой на значение элемента массива документов el_{v_1} :

$$\begin{aligned}
 T(V_0) = \{ & [T(\text{key}_{v_1}), \\
 & \{[T(\text{key}_{v_2}), \\
 & \quad \{T(v_{12}), \{[T(\text{key}_{v_{13}}), T(v_{14}), T(v_{15})]\}\}, \\
 & \quad T(v_6), \\
 & \quad \{[T(\text{key}_{v_7}), T(v_{16})]\}\}, \\
 & \quad T(v_8), T(\text{ref}_{v_2}(v_8)), T(v_{10}), T(\text{ref}_{v_2}(v_{10})), \\
 & \quad T(v_{17}), T(\text{ref}_{v_1}(v_{17})), T(v_{19})\}], \\
 & T(v_3), \\
 & T(\text{ref}_{v_2}(v_3))\}], \quad (2)
 \end{aligned}$$

Таблица 1

Дополнительные спецификации для отображения html-документа

Тип	Объект	Описание
HTMR (html root element)	$V^@, V^#, V^*$	Отмечает корневой элемент для иерархии html-элементов
HTML (html element tag)	$V^@, v: v \in V^@, T(v) = \{T(\text{tag}), \{[T(\text{key}_{\text{attrs}}), T(\text{attr}), T(\text{val})]\}\}$	Отображает структуру как html-элемента в виде $\langle \text{tag}[a_1 = v_1 \dots a_n = v_n] \rangle \dots \langle / \text{tag} \rangle$
NLB (no line break)	V^*	Отменяет переносы строк для терминальных вершин

всех гипертекстовых ссылок необходимо выбрать в массиве tags вершину с названием `href`, в котором находятся все элементы «а», имеющие атрибут «`href`». Для того чтобы получить набор гиперссылок с заданными значениями, необходимо выполнить соответствующий запрос, задав в качестве значения вершины v_{12} метку html-элемента: «а», вершины v_{14} название html-атрибута: «`href`» и вершины v_{15} значение атрибута «`href`». Структура результата запроса будет аналогична структуре элемента массива, соответствующего индексу по атрибуту v_{12} (названия элементов).

Среди гипертекстовых html-документов будем рассматривать только правильные html-документы в смысле [7]. Кроме того, важно отметить, что значения html-элементов «определяются только в контексте всего документа» [7]. Информацию о том, какие представления данных можно построить в частности на основе html-элементов, можно найти в работе [8]. Относительно гипервершинной модели данных [3] необходимо отметить, что в модели гипертекста на основе СУБД НИКА каждый элемент массива документов v_1 можно понимать как гипервершину, представляющую собой иерархию вершин el_{v_1} , которая связана с другими гипервершинами того же массива документов v_1 посредством гипертекстовых ссылок, т. е. ссылок на значения $gef_{v_1}(v_{17})$. Такая гипервершина может инкапсулировать метод отображения иерархии [8] для отображения html-документа, записанного в БД. Для отображения такого html-документа в программу html/xml-сервера СУБД НИКА [2] были добавлены дополнительные спецификации для отображения вершин БД (см. табл. 1).

Спецификации назначаются вершинам в описании данных [9] (рис. 2).

Перечень всех основных спецификаций отображения и их классификация описаны в [9]. Таким образом, на основании баз(ы) данных прикладной области мы получаем базу данных документов, имеющую структуру (2). Каждый документ, как отображение вершины исходной БД заданным методом отображения, имеет путь или координату в этой БД. Поскольку каждое описание данных определяет единственное отображение для каждой верши-

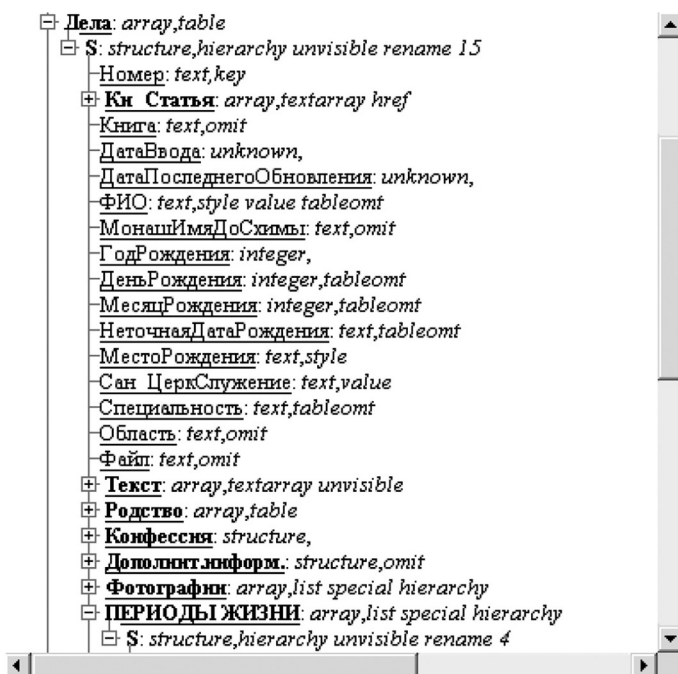


Рис. 2. Назначение спецификаций вершинам описания БД НИКА

ны, то название файла описания данных f (схемы данных) и координата вершины c_0 могут служить для идентификации документа. Назовем пару (f, c_0) идентификатором документа. С их помощью можно проверять наличие документа в базе данных при записи нового документа и осуществлять поиск документа по координате c_0 в исходной БД. Набор идентификаторов (f, c_0) описывается с помощью вершины верхнего уровня типа массив, имеющей следующую структуру:

$$T(\text{ids}) = \{[T(\text{key}_{\text{ids}}), T(\text{refp}(\text{ids})), T(v_{20}), T(\text{ref}_{v_1}(v_{20}))]\}, \quad (3)$$

где $\text{ids} \in V^\#$ обозначает массив идентификаторов документов; $\text{key}_{\text{ids}} \in V^*$ — ключ элемента массива ids , имеющий значение или названия файла описания данных или значение элемента пути c_0 в базе данных; $\text{refp}(\text{ids}) \in V^p$ — ссылочная вершина на шаблон массива ids ; v_{20} : $v_{20} \in V^*$ — номер или идентификатор html-документа в массиве v_1 ; v_{21} : $v_{21} \in V^f$, $v_{21} = \text{ref}_{v_1}(v_{19})$ — ссылочная вершина на значение, осуществляющая переход на заданный документ в массиве v_1 по ключу v_{19} . Описанная структура отображает множество идентификаторов документов (f, c_0) на множество внутренних

идентификаторов или номеров документов в БД. Определение (3) дает иерархическое представление идентификаторов (f, c_0) так, что координата в БД к документу представляет собой название схемы f и элементы пути c_0 , следующие в том же порядке, что и в c_0 (не считая корневой вершины V_0 и вершины ids). Номер или идентификатор документа v_{19} задается только у элемента массива, соответствующего последнему элементу пути c_0 :

$$(V_0, ids, f, c_{01}, c_{02}, \dots, c_{0n}, v_{19}) \quad (4)$$

— координата документа в БД с использованием идентификатора (f, c_0) . Вершина типа массива, заданная с помощью определения (3), может служить не только для идентификации документов, но, поскольку обычно путь в базе данных несет также и смысловую нагрузку, отображая иерархическую подчиненность объектов предметной области, и в этом смысле путь в БД (4) есть описание объекта c_{0n} посредством документа с внутренним номером v_{19} и его местоположение в иерархии объектов более высоких уровней, то вершина ids (3) может служить своего рода классификатором в базе данных документов, предоставляя к ним доступ через иерархию объектов предметной области.

В качестве примера рассмотрим объект «Дела» БД «Новомученики и исповедники Русской православной церкви XX в.» (см. <http://pstbi.ru>). Этот объект является массивом, элемент которого представляет собой сложную структуру для описания жизни пострадавших. Одним из подчиненных объектов является объект «ПЕРИОДЫ ЖИЗНИ», которому в свою очередь подчинен объект «Аресты», содержащий все аресты в указанный период жизни. В заданной иерархии объектов путь в БД будет иметь вид (для дела с номером «oni.10»):

$$(\text{«NIKA_ROOT»}, \text{«Дела»}, \text{«oni.10»}, \text{«ПЕРИОДЫ ЖИЗНИ»}, \text{«2»}, \text{«Аресты»}, \text{«1»}), \quad (5)$$

где '2' — номер периода жизни, '1' — порядковый номер в массиве 'Аресты'. Информативными здесь являются номер дела и названия объектов. Номеру периода жизни «2» соответствует период 1909–1930 годов, а номеру ареста «1» — арест 1928 года. Путь в БД (5) определяет документ, кратко описывающий арест протоиерея Николая Михайловича Боголюбова в 1928 году. Такая идентификация необходима в случае частичной упорядоченности, когда, например, нет точных дат, а только известные факты арестов в определенные периоды жизни. Для пользователя более удобен другой объект в БД — это индекс, построенный по вершине «ФИО»: («NIKA_ROOT», «INDEX», «ФИО»). С помощью этого пути в классификаторе можно найти документ, содержащий жизнеописание пострадавшего, по атрибуту «ФИО».

Объект INDEX.УчебЗаведение.«Ярославская Духовная Семинария» содержит список выпускников семинарии, пострадавших за веру в годы репрессий (в июне 2009 года в списке 55 человек, из них канонизированы 27).

2. Расширение ядра посредством языка XSL. Идентификация html/xml-элементов

С точки зрения прикладной области является важной возможность расширения отображения html-документа из БД НИКА посредством языка XSL [10], если рассмотреть html-документ как частный случай xml-документа и определить ссылку на соответствующий xsl-шаблон. Набор таких шаблонов документов можно также поместить в базе данных НИКА, определив, кроме массива документов, массив шаблонов документов. Каждый шаблон можно рассматривать как иерархию xsl-элементов аналогично иерархии html-элементов. Структуру html-документа e_{i-1} необходимо при этом дополнить ссылкой на значение на элемент массива xsl-шаблонов документов. Язык HTML [11], рассмотренный как подмножество языка XML, может быть расширен посредством языка XSL до языка разметки, определяющего содержательную структуру документа, совместив ее со спецификациями представления документа. Пусть задан xsl-шаблон, определяющий тождественное отображение исходного множества xml-элементов в то же самое множество xml-элементов [9]:

```
<xsl:transform>
<xsl:template match = "*">
<xsl:variable name = "tag" select = "name(current())"/>
<xsl:element name = "{$tag}">
<xsl:copy-of select = "attribute::*"/>
<xsl:apply-templates/>
</xsl:element>
</xsl:template>
</xsl:transform>
```

Для идентификации html-документов дополним каждый элемент атрибутом id, присваивая ему значение посредством элемента xsl:attribute и функции generate-id:

```
<xsl:attribute name = "id">
<xsl:value-of select = "generate-id(current())"/>
</xsl:attribute>
```

Другой способ идентификации — нумерация html-элементов в порядке левого обхода дерева элементов [7], используя, например, следующий шаблон для обхода дерева элементов [10]:

```
<xsl:transform>
<xsl:template match = "html">
<xsl:variable name = "contents">
```



```
<xsl:call-template name = "Tree-Left-Walk">
<xsl:with-param name = "level" select = "number(1)"/>
</xsl:call-template>
</xsl:variable>
<html>
<xsl:copy-of select = "$contents"/>
</html>
</xsl:template>

<xsl:template name = "Tree-Left-Walk">
<xsl:param name = "level"/>
<xsl:for-each select = "*/text()">
<xsl:variable name = "tag" select = "name(current())"/>
<xsl:choose>
<xsl:when test = "current()/* or current()/text()">
<xsl:element name = "{$tag}">
<xsl:attribute name = "id">
<xsl:value-of select = "number($level) + count(preceding:*)"/>
</xsl:attribute>
<xsl:copy-of select = "attribute::*"/>
<xsl:call-template name = "Tree-Left-Walk">
<xsl:with-param name = "level" select = "number($level) + 1"/>
</xsl:call-template>
</xsl:element>
</xsl:when>
<xsl:otherwise>
<xsl:value-of select = "current()"/>
</xsl:otherwise>
</xsl:choose>
</xsl:for-each>
</xsl:template>
</xsl:transform>
```

Способ идентификации посредством нумерации элементов в порядке левого обхода имеет преимущество в сравнении с присваиванием случайного идентификатора в том, что получается упорядоченный по идентификаторам набор элементов. Однако при использовании левого обхода задается порядок следования отображенных элементов такой же, как и для исходных элементов при последовательном копировании элементов в результирующее дерево, поэтому в общем случае лучше использовать шаблоны с применением элемента `xsl:apply-templates`, с помощью которого шаблоны имеют более ясный вид. При последовательном применении нескольких XSLT-преобразований (многоуровневое XSLT-преобразование) к xml-документу можно, например, первым определить шаблоны для преобразования в заданный набор элементов, вторым — осуществить нумерацию элементов в порядке левого обхода.

3. Модель системы в виде трехуровневой абстрактной гипертекстовой машины

Можно выделить три основных уровня в структуре гипертекстовой системы:

- пользовательский интерфейс (программная компонента обозревателя Интернета);
- гипертекстовая машина (http-сервер);
- база данных (СУБД НИКА).

Такая структура близка модели гипертекста в виде абстрактной гипертекстовой машины, описанной в [12]. В соответствии с этой моделью гипертекстовая машина является сервером общего назначения с использованием транзакций и работающей в многопользовательском режиме. Указанный сервер расположен между уровнями базы данных и пользовательским интерфейсом. Сервер запоминает всю информацию в граф или базу данных, управляет всей гипертекстовой информацией и связывается с пользовательским приложением через протокол управления потоком. В качестве указанного протокола http-сервер использует гипертекстовый интернет-протокол передачи. Уровень гипертекстовой машины — это уровень стандартизации формата импорта-экспорта для гипертекстовой системы, на котором определяются формат и свойства остальных уровней системы, чтобы обеспечить возможность передачи информации от одной гипертекстовой системы к другой. Обмен гипертекста включает кроме обмена данных в узлах также передачу ссылочной информации, что важно, когда существует возможность потери некоторой ссылочной информации.

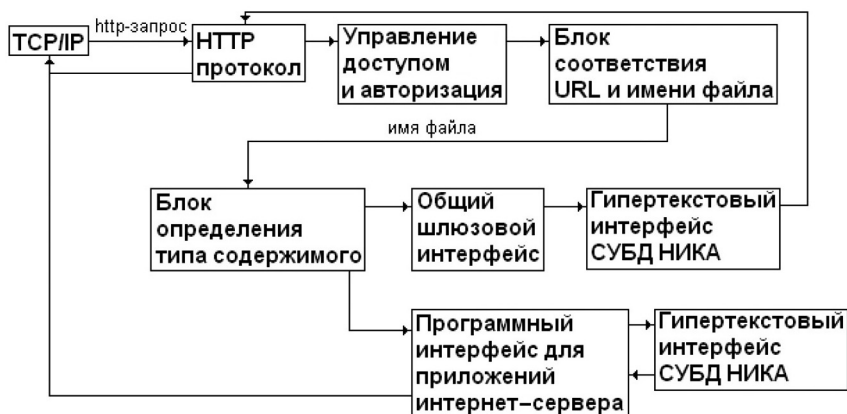


Рис. 3. Взаимосвязь СУБД НИКА и http-сервера

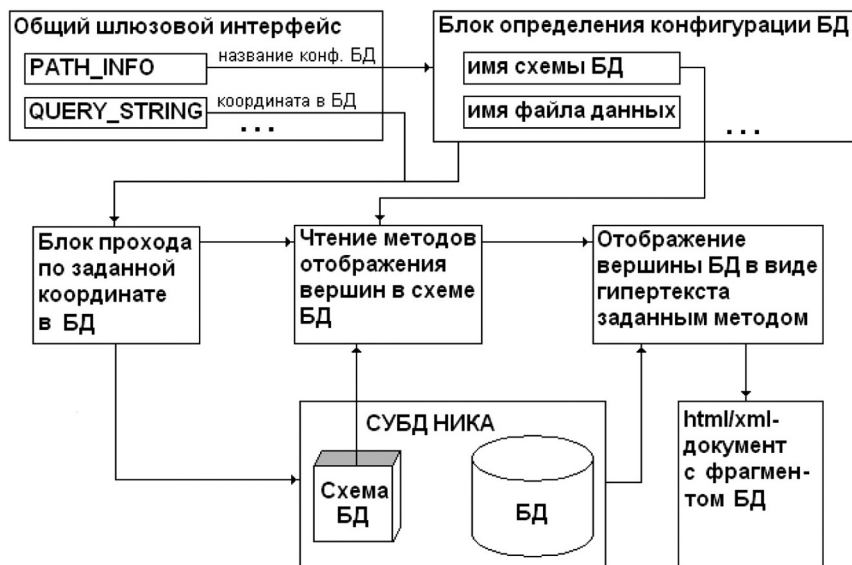


Рис. 4. Гипертекстовый интерфейс СУБД НИКА

Рассмотрим более подробно уровень базы данных. Использование СУБД НИКА как базы данных для гипертекстовой системы дает преимущество, как было отмечено выше, связанное с близостью моделей данных гипертекста и СУБД НИКА. Уровни http-сервера и СУБД НИКА связываются в соответствии со схемой, показанной на рис. 3.

Как видно из рис. 3, возможны два альтернативных способа связи СУБД НИКА с http-сервером — это общий шлюзовый интерфейс [2] и программный интерфейс для приложений интернет-сервера. Гипертекстовый интерфейс СУБД НИКА [2] представляет собой программный модуль, создающий гипертекстовые документы в форматах html и xml, содержащие запрашиваемые фрагменты базы данных, на основе спецификаций (методов) отображения вершин [9] (рис. 4).

На рис. 4 основным блоком интерфейса является процедура отображения вершины БД в виде гипертекста методом отображения, который определяется из схемы БД. Название БД задается переменной окружения PATH_INFO, а координата отображаемой вершины БД — переменной окружения QUERY_STRING. По названию БД в файлах конфигурации, представляющих собой БД НИКА, читаются параметры БД, включая имя файла схемы и имя файла данных. Результат отображения в виде html/xml-документа, содержащего фрагмент БД, передается http-серверу, который передает его по http-протоколу обозревателю Интернета. Процедура отображе-

ния реализует методы отображения [9]. Дополнительно к описанным возможностям реализована процедура записи html/xml-документов, содержащих избранные фрагменты исходной БД, в БД документов со схемой, изображенной на рис. 1. Объединив три уровня функционирования системы — уровень пользовательского интерфейса, уровень гипертекстовой машины и уровень БД (рис. 2) — в одну программу, мы получаем самостоятельную гипертекстовую систему на основе СУБД НИКА, которая включает в себя описанную процедуру записи в базу данных документов, также представленных в БД НИКА. Основное отличие от html/xml-документов, хранящихся в текстовом виде, состоит в том, что в БД html/xml-документы хранятся в виде семантических блоков, размеченных выделенными спецсимволами. Гипертекстовые ссылки также реализуются средствами СУБД НИКА.

Система, построенная таким способом, обладает возможностями, свойственными гипертексту, т. е. текст документов перестает быть замкнутым и становится открытым, в него можно вставлять новые фрагменты, указывая для них связи с имеющимися фрагментами, не нарушая структуры текста.

Использование html/xml-документов в их исходном формате совмещает «физическое» и «логическое» представление данных [14]. Применение СУБД НИКА [1] в качестве гипертекстовой системы с использованием HTML, XML, XSL технологий позволяет исключить регулярный разбор документов и ввести быструю навигацию по узлам БД, а также построить индекс по реквизитам всех объектов и подобъектов.

4. Методы обхода и отображения сложных структур данных

Назовем «семантической окрестностью» данного документа подмножество гипертекстовых документов, находящихся от данного документа в графе документов в пределах некоторого заданного расстояния (минимальный путь в графе). «Семантической близостью» двух документов назовем попадание одного из них в семантическую окрестность другого, а расстояние в графе между ними — степень этой близости, которая является критерием отбора документов при поиске. В СУБД НИКА существует поиск объектов БД по атрибутам, удовлетворяющим заданным условиям, полнотекстовый поиск, а также поиск семантических блоков по заданным условиям на атрибуты всех включающих и внутренних блоков [13].

В работе [15] выделяется пять общих методов обхода сложных структур данных, которые применимы и к гипертексту: прокрутка, просмотр, поиск, исследование, случайные передвижения. Эти методы используются во многих работах, например в [16, 17] и др. Приводятся графы обхода, соответствующие указанным методам. Графы были получены на основе изучения

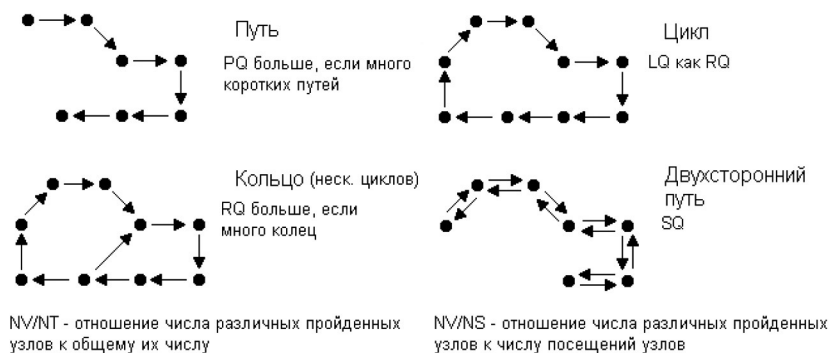


Рис. 5. Графическое представление четырех типов элементов обхода

статистических показателей моделей поиска информации пользователями при обходе базы данных. Они определяются следующими показателями: характеристика числа путей (PQ), характеристика числа колец (RQ), характеристика числа циклов (LQ), характеристика числа двухсторонних путей (SQ), отношение числа различных узлов, посещенных пользователями, к общему числу узлов (NV/NT), отношение числа различных узлов, посещенных пользователями, к общему числу посещений узлов (NV/NS). Здесь путь — это маршрут в графе БД (данные — узлы, дуги определяют связи между данными), который не проходит ни через какой узел дважды; кольцо — это маршрут, в котором начальная вершина совпадает с конечной, причем кольцо может включать другие кольца; цикл — это кольцо, которое не содержит других колец (простое кольцо); двухсторонний путь — это маршрут, проходящий через некоторые вершины и проходящий обратно через те же самые вершины (рис. 5).

Показатель PQ имеет высокий уровень, если много коротких путей, и низкий уровень, если небольшое количество длинных путей; RQ — высокий, если имеется множество колец, и низкий в противном случае; LQ аналогично RQ (как частный случай); SQ — обычно высокий для иерархических БД при обходе иерархии вглубь и обратном возврате; NV/NT — высокий в случае большего покрытия БД при ее обходе; NV/NS — низкий в случае высокой степени повторных посещений узлов. На рис. 6 указаны уровни показателей для различных методов просмотра.

Все эти методы реализованы в СУБД НИКА. Можно проиллюстрировать методы просмотра на примере БД «Новомученики и исповедники Русской православной церкви XX в.». Например, просмотр списка фамилий пострадавших или сводной таблицы, содержащей сан, имя, фамилию, год рождения, год кончины дает нам некоторый обзор базы данных и может быть отнесен к методу обхода «Прокрутка» БД, при этом покрывается большая область БД без прохода иерархии вглубь описаний подобъектов.

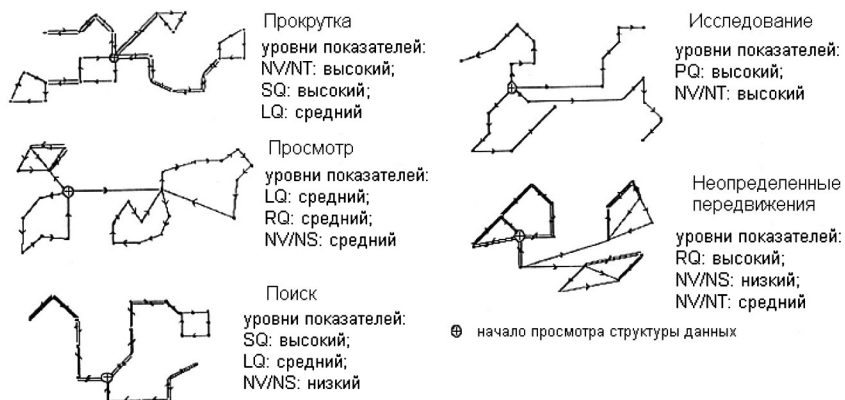


Рис. 6. Примеры графического представления пяти методов обхода сложных структур данных

Метод обхода «Просмотр» используется, когда необходимо, например, по фамилии перейти на сведения о пострадавшем, т. е. следование по пути до тех пор, пока не будет достигнута цель. Метод обхода «Поиск» используется при поиске точной цели и соответствует запросу к БД, когда ищутся объекты, имеющие заданные значения атрибутов, например выдать всех пострадавших, имеющих сан священника и служивших в Москве. Метод обхода «Исследование» используется для расширения знания о предметной области, при этом высокие показатели PQ и NV/NT, например просмотр оглавления или индекса БД на предмет изучения, из каких полей он состоит — ФИО, Год_Рождения, Сан и т. д., а также просмотр отдельных объектов БД. Наконец, метод обхода «Неопределенные передвижения» используется, когда нет определенной цели поиска, и обход данных осуществляется в случайном порядке и, возможно, с повторениями. Примером может служить многократный просмотр различных сведений о пострадавших. Кроме методов просмотра реализованы методы отображения в виде последовательности, таблицы, иерархии вершин или их комбинаций. В работе [8] рассматриваются эти методы и случаи их оптимальности.

5. Математическая модель гипертекста на основе СУБД НИКА

Описываемое применение СУБД НИКА как гипертекстовой системы является правомерным в виду двойственности структуры БД описания документа в виде иерархии html/xml-элементов (2) и структуры исходных html/xml-документов [18].

Исходный гипертекст может быть описан посредством следующей ссылочной модели [19]. Пусть заданы три множества B , S и D — базовых компонент, источников ссылок и приемников ссылок соответственно. Компонент $c = (b, S_c, D_c) \in (B \times 2^S \times 2^D)$ есть тройка, состоящая из базовой компоненты $b \in B$, подмножества $S_c \subseteq S$ источников ссылок и подмножества $D_c \subseteq D$ приемников ссылок. Множество $L \subseteq (C \times S) \times (C \times D)$, где каждое $l = ((c, s), (c', d)) \in L$, $s \in S_c$, $d \in D_c$, называется множеством ссылок. Структура

$$H = \langle C, L \rangle \quad (6)$$

называется гипертекстом. Ссылка $l = ((c, s_c), (c', d_c)) \in L$ связывает источник ссылки s_c компоненты c с приемником ссылки d_c компоненты c' .

Введем бинарное отношение $\rightarrow$ на множестве компонент C : $\rightarrow \subseteq C \times C$, описывающее связь между компонентами для гипертекста H . Для любых компонент $c, c' \in C$, $c \rightarrow c'$ тогда и только тогда, когда существует ссылка $l = ((c, s), (c', d)) \in L$. Зададим множество базовых компонент B посредством элементов массива документов v_1 , тип которого задается определением (2) для типа базы данных $T(V_0)$: $B = \text{dat}(v_1)$, где dat обозначает данные для вершины, указанной в качестве аргумента, v_1 определяется типом $T(v_1)$; множество источников ссылок S — посредством номеров документов key_{v_1} и идентификаторов key_{v_2} всех гипертекстовых элементов всех документов множества B : $S \subseteq I_d \times I_h$, где

$$\begin{aligned} I_d &= \{i_d \mid i_d = \text{dat}(\text{key}_{v_1}), \text{dat}(\text{el}_{v_1}) \in B\}, \\ I_h &= \{i_h \mid i_h = \text{dat}(\text{key}_{v_2}), \text{dat}(\text{el}_{v_2}) \in E_h, \text{dat}(\text{el}_{v_1}) \in B\}, \end{aligned}$$

E_h — множество всех гипертекстовых элементов, причем $(i_d, i_h) \in S$, i_h : $E_h \subset \text{dat}(v_2)$ для документа с номером i_d ; множество приемников ссылок D — посредством номеров документов key_{v_1} и типов всех гипертекстовых ссылок, задаваемых идентификаторами html -элементов v_{19} , всех документов множества B : $D \subseteq I_d \times I_t$, где

$$I_t = \{i_t \mid i_t = \text{dat}(v_{19}), \text{dat}(\text{el}_{v_2}) \in E_h, \text{dat}(\text{el}_{v_1}) \in B\},$$

причем $(i_d, i_t) \in D$, i_t : $E_h \subset \text{dat}(v_2)$ для документа с номером i_d . Таким образом, определены основные множества B , S и D в модели (6) с использованием определения типа БД [5] и реализации этого типа в БД НИКА (2). Следовательно, структура (2) соответствует гипертекстовой модели (6). Кроме того, при отображении исходных html/xml -документов в базу данных, описываемую структурой (2), сохраняется иерархия содержащихся в них элементов. Это следует из способа заполнения элементами массива v_2 в определении (2).

В рассмотренной модели гипертекста (6) не учитывается динамика работы с системой гипертекста. С использованием динамической струк-

туры строится модель гипертекста на основе сетей Петри. Указанная модель подробно рассматривается в [20]. В этой модели компонента сети Петри расширяется дополнительными компонентами, связанными с конкретной реализацией гипертекста. Рассмотрим гипертекст в виде сети Петри [21]:

$$H = \langle P, T, I, O, F, M, \delta \rangle, \quad (7)$$

где P — конечное множество позиций или мест сети ($P \neq \emptyset$); T — конечное множество переходов ($T \neq \emptyset$), $P \cap T = \emptyset$; F — множество дуг сети, $F \subset (P \times T) \cup (T \times P)$; $I(t, p)$ — кратность дуги из позиции в переход; $O(t, p)$ — кратность дуги из перехода в позицию; $M: P \rightarrow N$ — разметка сети, т. е. функция, которая ставит в соответствие каждой позиции целое неотрицательное число $M(p)$; $\delta(M, t_j) = M'$ — функция изменения состояния, вызванного срабатыванием перехода t_j (в случае если t_j разрешен при данной разметке M), $M, M' \in R(H)$, $R(H)$ — множество всех разметок сети H , достижимых от начальной разметки. Гипертекст в виде сети Петри (7) можно получить из исходного ориентированного графа следующим образом. Пусть граф гипертекста задан структурой (6), где C — вершины, представляющие собой гипертекстовые документы, L — дуги, представляющие собой гипертекстовые ссылки. Тогда указанному графу всегда можно однозначно сопоставить двудольный ориентированный граф, в котором множество вершин исходного графа C представлено множеством вершин P (позиции) и множество дуг L представлено множеством вершин T (переходы), множество дуг F связывает позиции и переходы в сеть так, что, если любые исходные вершины $c, c' \in C$ связаны дугой $l = ((c, s), (c', d)) \in L$, то вершинам c и c' будут соответствовать позиции $p, p' \in P$, дуге l переход $t \in T$ и p, t, p' связаны дугами $(p, t), (t, p') \in F$.

Кроме самой структуры гипертекста сеть Петри (7) отображает работу пользователей с гипертекстовой системой посредством разметки M и функции изменения состояния δ . Пользователь выбирает в документе определенную вершину БД, представляющую собой гипертекстовую ссылку. Это действие представляется в сети Петри для данного гипертекста срабатыванием соответствующего перехода t и новой разметкой $M' = \delta(M, t)$. В рассматриваемом случае переход t представляет координату в БД, реализующую выбранную гипертекстовую ссылку. При этом метка перемещается из позиции p в позицию p' , соответствующих исходному документу и документу, в который был осуществлен переход по ссылке, и содержащих фрагменты БД. Текущая разметка M определяет возможные срабатывания переходов. Кроме описанных переходов сети Петри гипертекста, можно определить множество T_u дополнительных переходов, которые вместе с уже существующими переходами между позициями-вершинами связывают все позиции друг с другом. Эти переходы соот-

ветствуют заданию ссылки в адресной строке обозревателя, а также обновлению текущего документа.

Все множество переходов связывает все позиции-вершины БД друг с другом и неявно задает ссылки с любого на любой уровень БД, которые возможно определить средствами и самой БД НИКА. Обозначим в сети Петри множество отображенных вершин базы данных и множество исходных иерархических и ссылочных связей между ними как V_{rd} и E_{rd} соответственно. Для описания работы гипертекстовой системы необходимо также добавить позицию p_n и дополнительные переходы t_{ni} , t_{ki} . Каждый переход t_{ni} связан с соответствующей позицией из множества V_{rd} . Позиция p_n есть некоторая начальная позиция, связанная со всеми переходами t_{ni} , $i = 1, \dots, |V_{rd}|$. Переходы t_{ki} обозначают окончание работы с гипертекстовым интерфейсом СУБД НИКА и с каждым t_{ki} связана соответствующая позиция-вершина из множества V_{rd} . Полученный из БД html/xml-документ может содержать также текущий путь (или координату) в БД, который определяет ссылки на вершины БД, начиная от корневой, которым подчинена данная вершина. Этим ссылкам соответствует подмножество переходов $T_{pi} \subset T_u$, $i = 1, \dots, |V_{rd}|$, которые связывают текущую позицию-вершину с позициями, соответствующими вершинам, которым подчинена данная.

Формально описанные позиции и переходы можно определить следующим образом:

$$P = V_{rd} \cup P_n, \text{ где } P_n = \{p_n\}, \\ T = E_{rd} \cup T_u \cup T_n \cup T_k, T_n = (\cup \{t_{ni}\}) \quad (i = 1, \dots, |V_{rd}|),$$

где $\cdot p_n = p_n = T_n$, $M_0(p_n) = 1$, $M_0(V_{rd}) = \mathbf{0}$, $\mathbf{0}$ — нулевой вектор размерности $|V_{rd}|$; $\forall t_{uij} \in T_u$, $\cdot t_{uij} = \{p_i\}$, $t_{uij} = \{p_j\}$, причем $p_i \in V_{rd}$, $p_j \in V_{rd} \setminus E(v_{i1\dots ik})$, где $i_1, \dots, i_{ik}$ — индексы вершин на соответствующих уровнях БД, начиная от корневой вершины, а $E(v)$ обозначает множество вершин БД, непосредственно подчиненных v ; $\cdot t_{ni} = \{p_n\}$, $t_{ni} = \{p_n, p_i\}$, $\cdot t_{ki} = \{p_i\}$, $t_{ki} = \emptyset$, $p_i \in V_{rd}$, $i = 1, \dots, |V_{rd}|$. В результате получается сеть Петри, все переходы которой обладают свойством: при любой достижимой в сети разметки M всегда существует такая достижимая от M разметка M' , что заданный переход может сработать. Это свойство называется свойством живости [21] любого перехода в сети, а значит и всей сети. В соответствии с теоремой об эквивалентности проблемы живости и проблемы достижимости в ней произвольной разметки получаем, что не любая разметка в построенной сети Петри достижима, т. е. не достижимы все разметки из множества D_i всех t -тупиковых разметок, для которых $M_i \leq M_{im}$, $M_i \in D_i$, $M_{im} \in D_{im}$, где D_{im} — конечное множество максимальных элементов множества D_i . В соответствии с теоремой о сводимости проблемы достижимости произвольной размет-

ки к проблеме достижимости нулевой разметки [21] получаем, что разметка $\mathbf{0} = (0, 0, \dots, 0)$ размерности $|P|$ не достижима. Это следует также из того, что позиция p_n всегда содержит одну метку, так как начальная разметка $M_0(p_n) = 1$ и множество входных и выходных переходов для позиции p_n совпадают $\cdot p_n = p_n \cdot = T_n$. Позиция p_n моделирует первоначальные обращения к БД по заданной координате, при этом метка из позиции p_n перемещается в позицию p_i , соответствующей заданной координате в БД, а также в соответствии с определением позиции p_n помещается другая метка в позицию p_n : $M_0[t_{ni} > M, M_0(p_n) = 1, M_0(p_i) = 0, M(p_n) = 1, M(p_i) = 1$, где символ « $[t >]$ » обозначает отношение непосредственного следования разметок при срабатывании перехода t . Таким способом может быть описано любое обращение к БД по заданной координате $M_1[t > M_2, M_1(p_i) \geq 1, M_1(p_j) \geq 0, M_2(p_i) = M_1(p_i) - 1, M_2(p_j) = M_1(p_j) + 1, p_i, p_j \in V, t \in E_{rd}, p_i \in \cdot t, p_j \in t$. Завершение работы с БД моделируется посредством срабатываний переходов t_{ki} . Срабатывание перехода t_{ki} приводит к удалению метки из соответствующей позиции p_i , если метка находится в позиции-вершине более заданного промежутка времени Δt_k : $M_1[t_{ki} > M_2, M_1(p_i) \geq 1, M_2(p_i) = M_1(p_i) - 1, p \neq p_n$. Необходимо отметить, что координата в БД в терминах сетей Петри определяет некоторую последовательность срабатывания переходов, связывающих вершины, входящие в заданную координату в БД. Такие последовательности образуют подмножество свободного языка сети $L(H)$.

Перестроим исходную сеть Петри H таким образом, чтобы ее язык выражался в виде последовательностей координат в БД. Для этого каждый переход t_{uij} из множества T_u заменим на последовательность чередующихся позиций и переходов, причем число переходов совпадает с числом вершин в пути в БД до заданной вершины v_j . Аналогично заменим переходы из множества T_n . В полученной сети H' пометим переходы посредством частичной помечающей функции $\Sigma: T \rightarrow A$, где A — алфавит, состоящий из имен нетерминальных вершин, не являющихся элементами массива, и ключей БД: $A = \& V \setminus (\& V^* \cup \& El_v) \cup \text{dat}(El_v)$ ($\& V$ обозначает множество имен вершин), так что переход t_i , являющийся «входным» для вершины v_i , помечается символом $\& v_i, i = 1, \dots, |V_{rd}|$. Также помечаются последовательности переходов, заменяющих переходы из множеств T_u и T_n (т. е. символами $\& v_0, \dots, \& v_i$). Остальные переходы помечаются «пустым» символом λ . Язык помеченной сети Петри с λ -переходами $L(H')$ будет состоять из слов, представляющих собой последовательности координат в БД. Таким образом, математическое описание [5] БД НИКА применимо к гипертекстовой системе, построенной на ее основе. Обобщая описание сети Петри для БД НИКА, отметим, что схема БД НИКА [4] определяет класс эквивалентных сетей Петри, тип которых задан посредством схемы БД в смысле [5].

Заключение

СУБД НИКА может быть использована как основа для построения справочной информационной системы с интернет-обозревателем в качестве программной компоненты. При этом текущие документы можно сохранить в БД избранных документов посредством описанного в данной статье метода. Таким образом можно сохранить гипертекстовые документы, найденные и отображенные одним или рядом запросов из одной или нескольких БД. При этом схемы БД, из которых выбираются документы, могут сколь угодно сильно отличаться.

Литература

1. Емельянов Н. Е. НИКА: гипертекстовая система или СУБД? // Базы данных и гипертекстовые системы. М., 1991. № 12. С. 4–9.
2. Emelyanov N. E., Muhanov I. V., Tishchenko V. A. Web-server on the basis of NIKA DBMS // Proc. of the third international workshop on advances in databases and information systems. ACM SIGMOD. Moscow, September 10–13 1996. Vol. 2. P. 58–59.
3. Levene M. On the information content of semi-structured databases // Acta cybernetica. 1998. Vol. 13. № 3. P. 257–276.
4. Годунов А., Емельянов Н., Космынин А., Солдатов В. Система НИКА // Системы управления базами данных и знаний. М: Финансы и статистика, 1991. С. 209–248
5. Abitebool S., Hull R. Restructuring hierarchical database objects // Theoretical computer science. 1988. Vol. 62. P. 3–38.
6. Emelyanov N., Godunov A., Romanov A. Indexing of objects in complex structures // Japan-CIS Symposium on knowledge based software engineering — JCKBSE-94. 1994. P. 63–66.
7. Богачева А. Н., Емельянов Н. Е. Семантическая модель документа // Системные исследования. Ежегодник 2001. М.: URSS, 2003. С. 360–375.
8. Емельянов Н. Е. Теоретический анализ документного интерфейса. Препринт. М.: ВНИИСИ, 1987. 38 с.
9. Емельянов Н. Е., Тищенко В. А. Методы отображения объектов для построения web-сервера объектно-ориентированной базы данных // Развитие безбумажных технологий в организационных системах / Сборник трудов ИСА РАН / Под ред. д. т. н. проф. В. Л. Арлазарова и д. т. н. проф. Н. Е. Емельянова. М.: URSS., 1999. С. 96–109.
10. Тищенко В. А. Применение языка XSL для отображения БД НИКА // Организационное управление и искусственный интеллект. Сборник трудов ИСА РАН / Под ред. д. т. н. проф. В. Л. Арлазарова и д. т. н. проф. Н. Е. Емельянова. М.: URSS. 2003. С. 149–175.
11. HTML 5. 2009. <http://dev.w3.org/html5/spec/Overview.html>.

12. *Campbell B., Goodman J. M.* A general purpose hypertext abstract machine // CACM. July 1988. Vol. 31. P. 7.
13. *Богданов А. С., Емельянов Н. Е., Ерохин В. И., Романов Б. Л.* Реализация запросной системы на основе XPath для ООСУБД НИКА // Организационное управление и искусственный интеллект. Сборник трудов ИСА РАН / Под ред. д. т. н. проф. В. Л. Арлазарова и д. т. н. проф. Н. Е. Емельянова. М.: URSS, 2003. С. 132–148.
14. *Коголовский М. Р.* Энциклопедия технологий баз данных. М.: Финансы и статистика. 2002. 326 с.
15. *Canter D., Rivers R., Storrs G.* Characterizing user navigation through complex data structures // Behaviour and information technology. 1985. Vol. 4. № 2. P. 93–102.
16. *Burkhard P., Lutz-Helmut S.* Internet work stratagies and their influence on learning outcome // Proc. of world conference on educational multimedia, hypermedia & telecommunications. 2002. P. 1598–1599.
17. *Levene M., Loizou G.* Web interaction and the navigation problem in hypertext // In: A. Kent, J. G. Williams, C. M. Hall (Eds.). Encyclopedia of Microcomputers. New York, N. Y.: Marcel Dekker, 2002. P. 381–398.
18. *Bogacheva A., Emelyanov N.* Duality between document structure and data base structure // Proceedings of the workshop on advances in database and information systems. May 1994. P. 83–89.
19. *Halasz F., Schwartz M.* The Dexter hypertext reference model // Communications of the ACM. 1994. Vol. 37 (2). P. 30–39.
20. *Scotts P. D., Furuta R.* Petri net based hypertext: document structure with browsing semantics // ACM transaction systems. Jan. 1989. Vol. 7 (1). P. 3–29.
21. *Кокин А. Г.* Сети Петри. Моделирование. Курган, 2005. 93 с.