УДК 515.142.33

РАЗРАБОТКА МОДУЛЕЙ СТРУКТУРИРОВАНИЯ ИНФОРМАЦИИ СИСТЕМЫ УПРАВЛЕНИЯ ДОКУМЕНТАМИ

М.В. СТЕРЖАНОВ, И.В. БАЙДАКОВ

Белорусский государственный университет информатики и радиоэлектроники
П. Бровки, 6, Минск, 220113, Беларусь

Поступила в редакцию 13 сентября 2012

Рассматривается система управления техническими документами (СУТД) Stagirites, имеющая ориентацию на автоматизацию работы с планово-конструкторской технической документацией. Рассматривается общая архитектура СУТД Stagirites и описываются модули создания и редактирования междокументных ссылок, примечаний, ключевых слов.

Ключевые слова: СУД, структурирование, навигация, ручное индексирование.

Введение

Среди актуальных задач современных информационных технологий особое место занимают проблемы разработки эффективных подходов к систематизации контента, которые затрагивают многие сферы человеческой жизнедеятельности, работа в которых основывается на информационных ресурсах. К таким областям можно отнести: Интернет, системы управления документами (СУД), библиотечные системы, образовательные системы и т.д.

СУД управляют небольшими взаимосвязанными единицами информации, и в данном контексте документ приобретает смысл гипертекста. Поскольку СУД управляют информацией, а у информации есть свой жизненный цикл, то, естественно, эти системы должны иметь адекватные средства управления контентом на каждом из этапов его жизни (создание, модификация, публикация, передача в архив и т.д.).

Описываемая в данной статье система управления техническими документами (СУТД) Stagirites ориентирована на автоматизацию работы с планово-конструкторской технической документацией. Очевидно, данная область вносит специфические требования к СУД. В процессе конструкторской и технологической подготовки производства появляется и используется большое количество документов, причем часть из них создается различными средствами конструкторской разработки, частично используется ранее разработанная документация (в бумажном и электронном видах). Работа над проектом осуществляется группой специалистов с различными ролями и обязанностями. При технологическом проектировании, а также оперативном планировании и управлении производственным процессом накапливается большой объем документов. Организация работы современного предприятия предполагает оперативное информационное обслуживание. В связи с вышеизложенным к системе предъявляются требования к обеспечению совместимости и согласованности данных, а также к скорости получения необходимой информации.

Под *структурированием* информации мы будем понимать расположение различных элементов информационного массива и создание между ними связей, обеспечивающих пользователям быстрый доступ к нужной информации.

Таким образом, важное значение для организации эффективного функционирования СУД имеют методы хранения и структурирования информации, навигации, поиска и фильтрации документов.

Общая архитектура системы

Описываемая СУТД построена с использованием традиционной распределенной многоуровневой архитектуры с «тонким» клиентом. Клиентским приложением пользователя является веб-браузер, обеспечивающий взаимодействие с системой через единый графический интерфейс. В качестве веб-сервера выступает приложение IBM HTTP Server, в которое встраивается разработанный нами модуль получения и переадресации запросов [1]. Клиент отсылает веб-серверу XML-запросы по протоколу http, которые затем транслируются на сервер приложений по протоколу TCP/IP. Сервер приложений реализует основную бизнес-логику взаимодействия объектов системы, выполняет запросы к серверу базы данных (СБД) и отсылает веб-серверу XML-ответы. СБД обслуживает базу данных (БД) и обеспечивает целостность и сохранность данных при их хранении, а также операциях ввода-вывода при доступе клиента к информации. Особенности построения презентационного уровня системы изложены в статье [2].

Опишем основную функциональность предлагаемой системы. Система представляет контент в виде множества отдельных записей. Записи упорядочиваются в каталоге с помощью иерархической древовидной структуры, называемой деревом публикаций. Выбор такой структуры представления данных является не случайным. Одним из центральных принципов работы с информацией является принцип модульности. В соответствии с этим принципом контент структурируется в виде отдельных блоков (публикаций и заголовков). Текстовые записи бывают трех типов: публикация, заголовок, статья. Запись, соответствующая корневому узлу дерева, называется публикацией. Каждая публикация содержит набор логически сгруппированных документов и может представлять отдельный проект, дело, книгу и т.п. Некорневые записи-контейнеры называются заголовками. Заголовками являются структурные элементы проектной документации, описывающие отдельные разделы. Статьи являются листьями дерева публикаций и хранят в себе текст, таблицы, изображения. Контент системы может создаваться пользователями вручную при помощи встроенного визуального WYSIWYG редактора, обладающего богатыми возможностями для редактирования текста и таблиц. Также имеется возможность автоматического импорта документов, представленных в формате MS Word, HTML, XML. Каждая запись содержит следующую метаинформацию: дату создания, имя создателя, дату последней модификации и имя пользователя, который осуществил последнюю модификацию записи.

Дерево публикаций обеспечивает целостность представления информации. Дерево является гибкой структурой, содержание каждого проекта может легко дополняться или изменяться. Помимо очевидных операций над узлами (добавление, удаление, копирование, перемещение) реализована операция «повторного использования» контента, которая помогает ликвидировать дублирование информации и позволяет строить новые узлы контента по уже имеющимся. При этом информация репозитория разделяется между узлами дерева. Данный принцип аналогичен концепции символьной ссылки в UNIX.

Основные сущности модели БД системы представлены на рисунке.

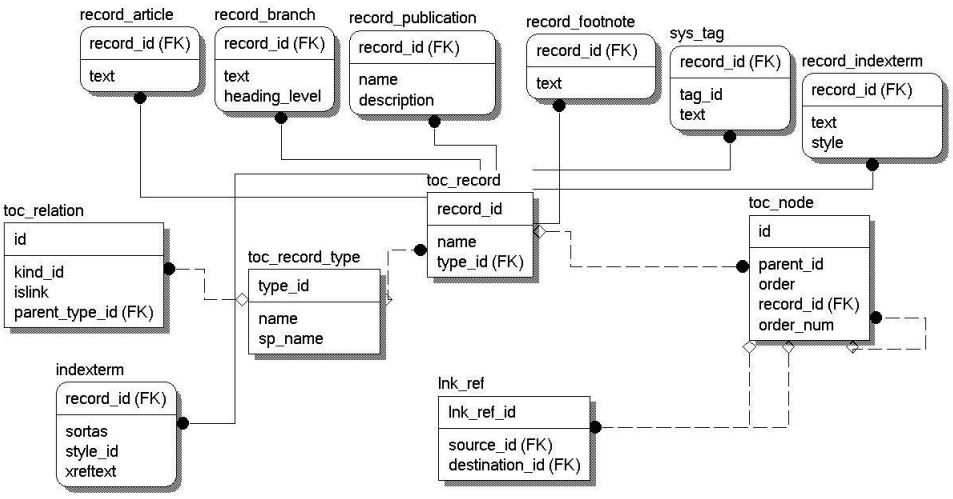

Рис. 1. Таблицы БД, описывающие основные сущности системы

Для каждого объекта в системе существует запись в таблице *toc_record*, которая служит для хранения дескриптора объекта и имени записи, отображаемой в дереве публикаций. При помощи внешнего ключа на таблицу *toc_record_type* можно узнать тип записи (элемент дерева публикаций, системный тег) и определить таблицу, содержащую непосредственно данные записей этого типа. Данное отношение имеет вид «один к одному» и реализуется посредством миграции первичного ключа *id* таблицы *toc_record* в качестве внешнего первичного ключа в другие таблицы. Например, текст публикаций хранится в таблице *record_publication*, текст заголовков − в таблице *record_branch*, текст статей − в таблице *record_article* (контент хранится в формате XHTML).

Таблица *toc_node* служит для структурирования объектов системы и представления связей между записями таблицы *toc_record* в виде отношения «родитель-потомок». Иерархия узлов дерева публикаций хранится именно в этой таблице. Каждый узел имеет уникальный идентификатор, связь с родителем (*parent_id*), а также порядковый номер, соответствующий позиции размещения в дереве по отношению к смежным узлам.

Таблица *toc_relation* позволяет задать для каждого типа узла разрешенные типы дочерних объектов. Например, для публикации в качестве дочерних узлов могут выступать только заголовки и статьи. С помощью таблицы *toc_filter* можно разрешить отображение в дереве публикаций только записей определенного типа. Например, компонент, представляющий дерево публикаций, может быть использован для отображения не только структуры документов, но и конфигурационных параметров системы. Данные два представления должны быть доступны разным пользователям. Поэтому для каждого пользователя дерево будет строиться с применением необходимого фильтра.

Введем понятие системного тега. Под системным тегом будем понимать элемент XHTML кода текстовой записи системы, имеющий имя *stag*. Системные теги служат для представления различных служебных и вспомогательных элементов контента и реализуются в виде HTML компонентов. Для каждого системного тега существует запись в таблице toc_record. Возможны следующие типы системных тегов: междокументная ссылка, примечание, индекс. Для хранения информации о системном теге служит таблица *stg_tag*. Поле *stg_tag.tag_id* является уникальным идентификатором тега и соответствует атрибуту *id* XML-представления тега. Атрибут *id* также является уникальным в системе. При копировании документа целиком, либо части документа, содержащей в себе системный тег, создается новый системный тег с отличным от исходного идентификатором *id*.

Средства структурирования информации

Удобство навигации является важной составляющей при работе с документами. Навигация является центральным понятием концепции гипертекста и означает управление процессом перемещения в гиперпространстве из произвольного узла отправления в узел прибытия [3]. Технически навигация осуществляется путем нажатия мышью на графически выделенные на экране компьютера объекты. Для обеспечения навигации записи могут быть связаны друг с другом с помощью междокументных ссылок (в дальнейшем просто ссылок). Ссылкой является специальная метка, назначением которой может быть как документ целиком, так и отдельная фраза в тексте. Ссылка может быть установлена на любую запись в системе, что обеспечивает создание связей между документами, находящимися в любом месте репозитория. В XHTML коде текстовой записи ссылка размещается в виде системного тега <stag:xref id={xref_GUID}></stag:xref>. Система отображает ссылки на записи разного уровня вложенности при помощи разного графического представления.

В базе данных механизм ссылок поддерживается при помощи таблицы *lnk_ref*. Поля *source_id* и *destination_id* представляют идентификаторы узла, содержащего запись, в которой имеется ссылка, и узла, содержащего запись, на которую указывает ссылка, соответственно. Опишем механизм работы ссылки. Идентификатор ссылки (xref_GUID) хранится в таблице *sys_tag* в поле *tag_id*. Обозначим запись таблицы *sys_tag*, содержащую код ссылки *R*, через *RS*. По значению поля id таблицы *sys_tag* можно найти запись *RL* таблицы *toc_record*, для которой справедливо равенство *RS.id= RL.record_id*. Данная запись *RL* в таблице *toc_record_type* имеет тип «ссылка». В таблице *toc_node* для записи *RL* имеется узел *NS*, значение которого равно

значению lnk_ref.source_id. Покажем способ нахождения назначения ссылки. По значению lnk_ref.destination_id находится запись *ND* в таблице toc_node. Для узла *ND* находим запись в таблице toc_record *NR*. Если запись *NR* имеет тип «текстовый документ», то ссылка *R* указывает на узел *ND*. Если запись *NR* имеет тип «якорь», то в таблице sys_tag находим запись *AR*, для которой выполняется *AR.id = NR*.record_id. Для записи *AR* найдем запись в таблице toc_node *AN*. Для узла *AN* найдем родительский узел *PN* при помощи значения поля *parent_id*. Узел *PN* является назначением ссылки *R*. Запись *PR*, соответствующая узлу *PN*, содержит в себе код < *stag*:*xanchor id*={*ANCHOR_GUID*}></*stag*:*xanchor*>.

«Битой» ссылкой называют такую ссылку, которая ссылается на отсутствующий по каким-либо причинам объект, например, если целевой документ был удален. При выборе такой ссылки система показывает окно с описанием ошибки. В системе реализован модуль нахождения «битых» ссылок, которые должны отсутствовать в опубликованном контенте.

Примечание − заметка, добавленная автором или рецензентом в документ. Примечания используют для комментирования документов, добавления к ним замечаний, предложений, рекомендаций и т.д. При этом содержание самого документа остается неизменным. Значки примечаний в тексте статьи отображаются арабскими цифрами. При нажатии на значок примечания система отображает текст примечаний, относящихся к данной статье, в отдельном окне внизу области просмотра и редактирования контента. При печати документа текст примечаний размещается внизу страницы. При удалении или добавлении нового примечания система автоматически перенумеровывает номера примечаний текущей статьи для поддержания последовательности нумерации. Примечания реализуются в виде системных тегов <*cms*:*note*> . Текст примечания хранится в таблице *record_footnote* и может содержать ссылки, а также быть форматированным различными стилями.

При работе с научной, технической литературой часто возникает необходимость ссылаться на различные источники. Формат и типы источников могут быть различными: ГОСТы, СНиПы, СанПины, другая литература. Однако само цитирование должно выполняться одинаково и с использованием строгих формальных правил. Нами предлагается специальный компонент, хранящий коллекции объектов цитирования. Пользователи СУД Stagirites могут редактировать списки объектов цитирования и ссылаться на них в тексте. Добавленная в текст цитата отображается специальным стилем. Ссылка на объект цитирования состоит из двух частей: цитаты в тексте и объекта цитирования. Объект цитирования содержит информацию, описывающую то, на что ссылаются (СанПин, ГОСТ, книга). Объект цитирования может быть использован многократно в одном документе. В тексте документа цитата представляется специальным тегом <*cms*:*citation*>. Существуют различные типы объектов цитирования. Каждый тип представляет документы и источники определенной предметной области и формата (например: правовые документы, научные статьи). Следовательно, с типом ассоциирован формат хранения и представления информации. Каждый объект цитирования принадлежит к определенной логической схеме. Схема описывает атрибуты объекта цитирования, их формат. Для схем существует механизм генерализации: возможно наследование полей схемы-родителя и расширение дополнительными полями схемы-потомка.

Индексированием называется процесс выбора терминов, наиболее точно отражающих содержимое документа. Система осуществляет индексирование контента в двух режимах: ручном и автоматическом. Автоматическое индексирование выполняется посредством индексатора, описанного в статье [4]. Ввод информации в систему сопровождается классификацией документов путем задания атрибутов и ключевых слов, аннотированием их содержания. Ручное индексирование информации выполняется с помощью деревьев ключевых слов. Листья данного дерева являются ссылками на индексированный контент и позволяют легко переходить к нужной информации. При создании индекса пользователь выбирает фразу в тексте и задает имя индекса, который будет на нее указывать. Дерево ключевых слов также может содержать внутренние ссылки. Т.е. одно место в тексте может быть индексировано из разных узлов дерева ключевых слов. Во избежание дублирования информации система позволяет создать индекс, ссылающийся на другой индекс. Например, индекс «дымовые трубы» может ссылаться на индекс «вентиляционная шахта», который указывает на соответствующий раздел документа ЕСКД. Каждой публикации может быть поставлено в соответствие несколько таких деревьев, представляющих иерархию ключевых слов и индексирующих документы, принадлежащие

данной публикации. Система позволяет выполнить поиск ключевых слов, что ускоряет процесс нахождения требуемой информации. Индекс может указывать как на документ целиком, так и на отдельную фразу в тексте. В XHTML коде документа индекс представлен системным тегом <*cms*:*index*>. При нажатии на значок проиндексированного текста система отображает соответствующий узел дерева ключевых слов. В БД дерево ключевых слов представлено в таблице *toc_node*, хранящей записи *toc_record* типа «indexterm». Корень дерева имеет тип «indexroot» и является дочерним узлом по отношению к узлу публикации. Важной особенностью использования дерева ключевых слов является то, что на основании хранимой в нем информации осуществляется построение предметного указателя при доставке контента. Аналогично механизму нахождения «битых» ссылок в системе реализован модуль валидации ключевых слов, который гарантирует, что все индексы ссылаются на существующие документы данной публикации. Пример дерева ключевых слов приведен на рисунке.

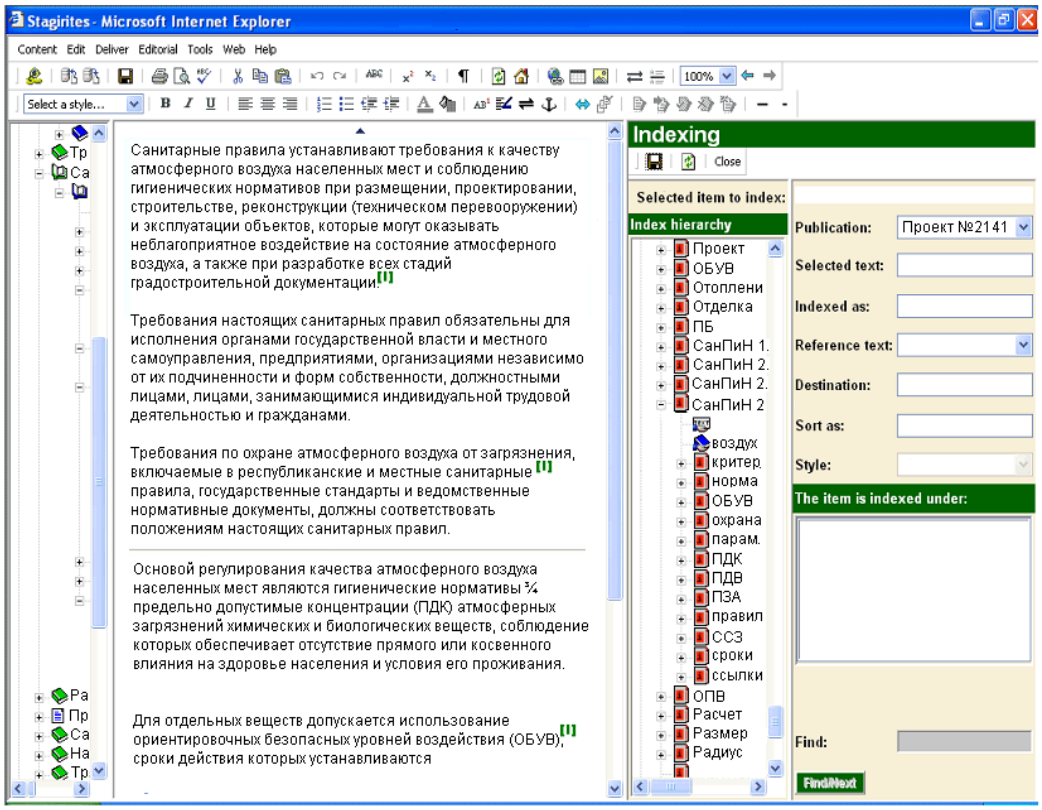

Рис. 2. Дерево ключевых слов

Важной характеристикой, описываемой СУТД является реализация возможности многократного использования информации. Система может преобразовывать контент в формат HTML для публикации на веб-сайте, а также в CD ROM формат в виде настольной БД. Изменение формата представления данных не нарушает созданных в СУТД отношений структуризации информации.

Заключение

Описанные в данной статье модули увеличивают производительность оператора за счет сокращения времени и усилий на нахождение требуемого материала. Применение модулей структурирования информации ведет к уменьшению количества обрабатываемой информации, что положительно скажется на работоспособности оператора и позволит выделить дополнительное время для анализа и принятия управленческих решений. Важной особенностью предложенных компонентов является возможность использования полученного логически-структурного содержания информации в различных приложениях и контекстах при доставке информации.

DEVELOPING DATA STRUCTURING MODULES FOR THE CONTENT MANAGEMENT SYSTEM

M.V. STERJANOV, I.V. BAIDACKOV

Abstract

General architecture of engineering document management system «Stagirites» is depicted in this article. The system provides engineers with a secure, collaborative web-based environment to create, capture, review, and manage, both completed and work-in-progress engineering documents. Modules for creating and editing internal links, footnotes, keywords are also described.

Список литературы

1. *Стержанов М.В., Байдаков И.В.* // Докл. БГУИР. 2012. № 4 (66). С. 62–67.
2. *Стержанов М.В., Байдаков И.В.* // Электроника Инфо. 2012. № 5. С. 110–112.
3. *Эпштейн В Л.* // РАН. Институт проблем управления. М., 1998.
4. *Стержанов М.В.* // Докл. БГУИР. 2012. № 6 (68). С. 95–99.