

ТЕХНОЛОГИЯ ПОСТРОЕНИЯ КОРПОРАТИВНОГО WEB-САЙТА

Рассматриваются основные требования к современному корпоративному сайту, предлагается подход к организации структуры сайта и технические подходы к его построению.

Одной из задач при построении автоматизированной корпоративной информационной системы является построение интернет-представительства, или Web-сайта.

Идея предлагаемой технологии заключается не только в том, чтобы создать представительство организации в сети Интернет, содержащее статичную часть, но и тесно интегрировать его функции с остальными составляющими информационной системы, предоставить возможность динамического изменения структуры, степени доступности и информационной составляющей в соответствии с нуждами организации.

ТРЕБОВАНИЯ К СОВРЕМЕННЫМ САЙТАМ

К современным корпоративным сайтам предъявляются следующие требования:

- единое внешнее оформление для всех генерируемых страниц (цветовая гамма, символика организации, пространственное расположение элементов страниц);
- единая и гибкая структура сайта;
- единообразие навигации по сайту, наличие карты сайта;
- удобочитаемость и интуитивная ясность генерируемых ссылок;
- минимальные затраты на поддержку, администрирование и развитие сайта;
- наличие внешней (пользовательской) и внутренней (корпоративной) частей сайта;
- возможность заполнения страниц с использованием Web-браузера, минимальные требования к аппаратно-программной оснащённости рабочего места;
- возможность посещения сайта с помощью всего множества существующих браузеров;
- наличие гибко настраиваемой системы безопасности (аутентификация пользователей, авторизация доступа к структурным и информационным элементам, возможность работы в защищенных сеансах (SSL, PKI));
- возможность оценки и контроля степени актуальности предоставляемой информации (отображение даты последнего изменения и ответственного за раздел);
- распределенность источников информации;
- тесная интеграция с существующими средствами автоматизации делопроизводства и учета;
- открытость встроенных программных средств, возможность обращения к открытым программным средствам сторонних производителей (технологии Java VM, COM, CORBA);
- возможность функционирования сайта в кластерном режиме (масштабируемость, отказоустойчивость и распределение нагрузки);
- возможность полнотекстового поиска по выбранным разделам или по сайту в целом;
- высокая производительность обслуживания пользователей.

ИНСТРУМЕНТАРИЙ

На текущий момент на рынке корпоративных систем, включающих поддержку служб Интернета, представлено множество продуктов и надстроек над существующими СУБД:

- Novell exteNd;
- Netscape FastTrack Server, Netscape Enterprise;
- Microsoft Internet Information Server, .NET;
- Oracle 9i;
- Apache Web Server;
- IBM Lotus Domino.

Все эти продукты имеют специфическую ориентацию, но с точки зрения построения Web-сайтов их можно отнести к одному классу.

Клиент-серверная среда Lotus Domino/Notes, лидер среди группового программного обеспечения, ориентирована, в общем случае, на приложения для автоматизации документооборота и деловых процессов [1].

Эта платформа включает в себя:

- интегрированную среду (Integrated Development Environment) для быстрой разработки приложений (Rapid Application Development), позволяющую быстро разрабатывать и разворачивать приложения и поддерживающую открытые интерфейсы COM, CORBA, языки Java и JavaScript;
- механизм наследования программных элементов (форм, подформ, видов, агентов и т.д.), позволяющий повторное использование разработанных структурных элементов и автоматическое их наследование, что решает задачу построения единого пользовательского интерфейса и единого внешнего оформления;
- встроенные средства репликации, позволяющие создавать распределенные источники информации с автоматической актуализацией данных;
- встроенную службу HTTP для генерирования страниц, соответствующих объектам Domino, включающую виртуальную машину Java (JVM);
- встроенный механизм индексирования информации, содержащейся в документах для последующего полнотекстового поиска;
- встроенную систему безопасности, использующую симметричные и асимметричные алгоритмы шифрования и электронной подписи, протоколы аутентификации (SSL) и поддерживающую инфраструктуру PKI;
- надстройки, позволяющие интегрировать разработанные приложения с реляционными источниками данных;
- службы для создания кластеров уровня приложений Domino и интернет-кластеров.

Таким образом, выбор в качестве основы для построения Web-сайта Lotus Domino удовлетворяет большинству из предъявляемых требований.

Остальные требования удовлетворяются технологией построения сайта с использованием встроенных средств разработки и международных стандартов Интернета.

Инструментами при разработке Web-сайта в этой среде являются:

- виды с критерием отбора документов и сортировкой в колонках;
- формы и подформы, являющиеся внешней оболочкой для документов и несущие функциональные составляющие интерфейса пользователя;
- агенты, запускающиеся по команде из формы, либо при наступлении определенного события (изменение документов в базе, сохранение Web-документа);
- язык @-формул, являющийся наиболее производительным по сравнению с остальными языками;
- язык Lotus Script версии 3.1, менее производительный, чем язык @-формул, но более гибкий в случае, когда необходимо произвести достаточно сложные действия над большим количеством документов;
- язык разметки гипертекста HTML и таблицы каскадных стилей CSS не являются встроенными средствами Domino, но их можно включать в генерируемые страницы;
- язык JavaScript, позволяющий создавать интерактивные приложения для браузера и клиента Notes [2].

ЛОГИЧЕСКАЯ СТРУКТУРА САЙТА

Логическая структура сайта в основном определяется организационной структурой и сферами деятельности организации. В большинстве случаев ее утверждает руководитель организации.

Существуют определенные требования к построению структур, обеспечивающие наглядное представление элементов, порядка их следования друг за другом и вложенности.

Одно из таких требований – «7±2» – задает количество элементов на уровне. Естественно, существуют и исключения из этого правила (к примеру, в разд. «Подразделения организации» может быть 10 и более подразделений).

Другое требование продиктовано техническими средствами просмотра сайтов. Это «правило трех кликов», требующее, чтобы до необходимой пользователю информации он мог «добраться», нажав не более трех раз на элемент системы навигации.

Оба эти требования ограничивают количество элементов на уровне, а также высоту дерева рубрикации.

Формально логическую структуру сайта можно представить либо в виде дерева (с неявным корневым элементом «стартовая страница»), либо в виде леса, деревьями в котором являются рубрики сайта первого уровня.

Для иллюстрации логической структуры и ее реализации в виде структурных элементов будем использовать следующий фрагмент структуры сайта:

1. Общие сведения
 - 1.1. Приветствие ректора
 - 1.2. Устав ТГУ
 - 1.3. Ученый совет ТГУ
 - 1.3.1. Состав
 - 1.3.2. Комиссии
 - 1.3.3. Решения
 - 1.3.4. Объявления о заседаниях
 - ...
 - 1.4. Ректорат
 - 1.5. Структура университета

- ...
2. Абитуриенту
 - 2.1. Факультеты, специальности
 - 2.2. Уровни образования и формы обучения
 - 2.3. Правила приема
 - 2.4. Вступительные экзамены
- ...

Для представления логической структуры в архитектуре сайта предусмотрены соответствующие «элементы структуры», исполненные в виде документов, создаваемых на основе формы «Элемент структуры».

АРХИТЕКТУРА САЙТА

Основу архитектуры сайта составляет главная база данных (core.nsf), содержащая как логическую структуру сайта, так и архитектурные элементы, общие для всех страниц сайта (рис. 1).

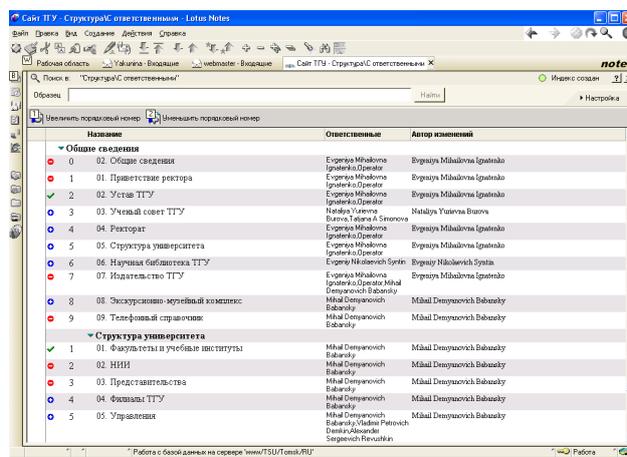


Рис. 1. Логическая структура сайта

Все остальные базы сайта являются специфичными для конкретных подразделений (информационно-рекламный отдел, ученый совет и др.), используют некоторые общие элементы (на основе механизма наследования дизайна) и подключаются к этой главной базе регистрацией в ней соответствующих ссылок.

Структура сайта представлена в виде иерархии документов с максимальной степенью вложенности 4 (что, вообще говоря, продиктовано существующей структурой сайта и удобством навигации, нежели программными ограничениями).

Как уже было сказано выше, элементы структуры служат для построения иерархии разделов сайта и автоматического создания удобочитаемых ссылок. Кроме того, сам структурный элемент может содержать и конечную информацию, отображаемую посетителю сайта в браузере.

Далее рассмотрим внутреннюю реализацию элемента структуры.

Поля элемента структуры

Форма структурного элемента (рис.2) содержит поля:
- **Имя документа.** Это имя будет отображаться в заголовке документа и будет включено в тег <title>.

- **Уровень документа.** Это многозначное поле, содержащее последовательный список имен документов родителей – путь от корня сайта до этого документа.

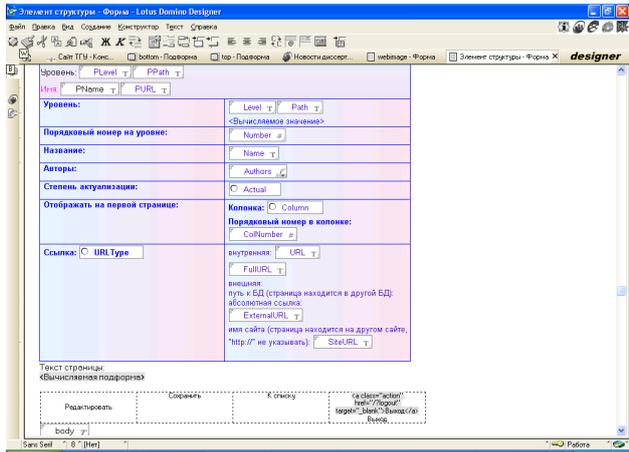


Рис. 2. Форма структурного элемента

Например, уровень у страницы «Объявления о заседании» содержит «Общие сведения :: Ученый совет ТГУ».

- Имя внутренней ссылки. Обязательное поле, состоящее из латинских букв и (при необходимости) арабских цифр, для поиска этого документа среди других документов на своем уровне иерархии. Например, для элемента «Объявления о заседаниях» на уровне common_sci_council внутренней ссылкой будет являться «announces».

При отображении списков документов в браузере Domino генерируют неудобочитаемые ссылки типа: <http://ic1004.ic.tsu.ru/webdesign/tsu/core.nsf/31D5942888B0C3E1C6256CC200499D2A/B5802D3C02007CE6C6256CC400452A93?OpenDocument>. Этот способ формирования внутренней ссылки не только исполняет поисковую функцию, но и позволяет решить проблему неудобочитаемости [3]. Полная ссылка будет выглядеть следующим образом:

http://ic1004.ic.tsu.ru/WebDesign/TSU/core.nsf/structurl/common_sci_council_announce.

- Тип внешней ссылки:

абсолютная: страница находится в текущей базе данных (core);

относительная:

1) если страница находится в другой (отличной от core) базе данных, дополнительно указывается путь к этой базе данных;

2) если, более того, страница находится на другом (отличном от текущего) сайте, тогда при регистрации дополнительно указывается имя этого сайта.

- Полная внутренняя ссылка. Эта ссылка состоит из последовательно располагающихся имен внутренних ссылок родительских документов, разделенных символом подчеркика «_». Например, страница, содержащая объявления о заседаниях ученого совета, имеет полную внутреннюю ссылку common_sci_council_announces.

- Порядковый номер на уровне. Уникальность документов структуры в пределах всей сердцевинной базы данных (и, следовательно, в пределах всего сайта) обеспечена требованием уникальности документов на каждом уровне и способом построения полных внутренних ссылок. Порядок же расположения, вне зависимости от лексикографического порядка названий страниц и внутренних ссылок, определяется от-

дельным полем — порядковый номер на уровне. Именно по значению в этом поле будет осуществляться сортировка элементов структуры (рис. 3). Если это поле не заполнено, то при сортировке вместо его значения подставляется значение 99, сдвигая страницу в самый низ относительно других.

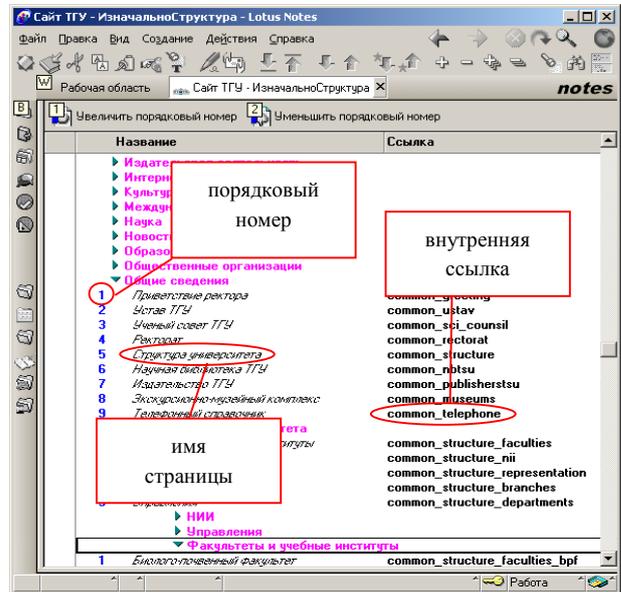


Рис. 3. Элементы структуры сайта

- Отображение на первой странице. Для создания динамической первой страницы в каждом элементе структуры задается, в которой из двух колонок будет видно название и ссылка на этот элемент. Такой способ организации первой страницы напоминает карту сайта, если большинство структурных элементов отображены на ней. Положение элементов в пределах колонки задается структурными и ссылочными полями, указанными выше.

- Авторы. Это поле с разделяемым доступом аутентифицированных пользователей к документам. Все пользователи, перечисленные в этом поле, имеют право изменять этот элемент структуры.

- Автор. Отдельное поле, хранящее имя пользователя, редактировавшего документ последним (рис. 4).

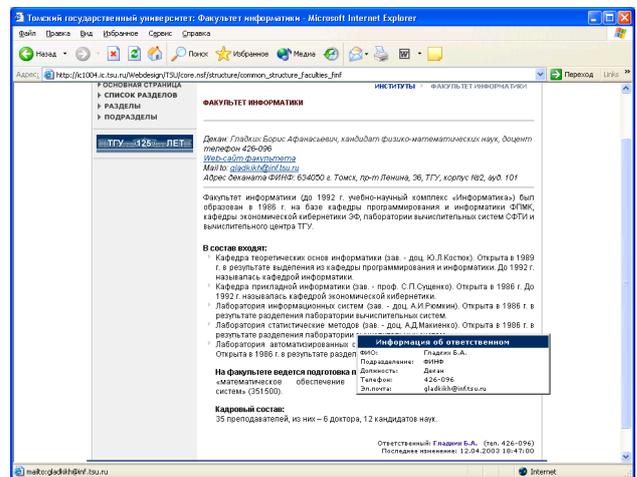


Рис. 4. Отображение контактной информации ответственного в Интернете

На основе этого поля и документа информации об ответственном на странице выводится информация о дате и времени последнего изменения и ФИО, структурное подразделение, должность и контактная информация ответственного.

- **Читатели.** Поле для распределения доступа на чтение документов аутентифицированными пользователями. Это поле содержит в обязательном порядке анонимного пользователя (Anonymous), чтобы обычный (не аутентифицированный) пользователь сайта мог получить доступ к документу.

- **Степень актуальности.** Редактируемое поле для корректора сайта или контролирующего наполнение сайта. Оно может содержать одно из трех значений:

1) *не заполнено.* Низшая степень актуальности. Страница должна быть немедленно заполнена либо удалена, если, конечно, этот структурный элемент не описывает внешнюю ссылку и по логике не должен содержать тела страницы;

2) *заполнено.* Средняя степень актуальности. Страница содержит определенную информацию, но нельзя говорить о ее актуальности и непротиворечивости;

3) *заполнено и актуально.* Информация была заполнена исполнителем и проверена корректором.

Это поле на страницах не отображается никоим образом, оно используется только в автоматизированном рабочем месте корректора.

- **Тело страницы.** Это основное форматированное поле, которое содержит всю информацию страницы. Это поле может содержать как обычный текст, так и форматированный текст, таблицы, изображения, ссылки и объекты. Это поле и будет доступно интернет-браузеру.

Подформы

Упомянутая выше возможность наследовать элементы форм была использована при создании подформ. Созданные подформы верхней и нижней частей отображаемых страниц были объявлены как наследуемые всеми другими базами данных.

Количество подформ, включаемых в формы, определяет, с одной стороны, гибкость программирования формы и высокую объектную степень абстракции, с другой стороны, производительность приложения в целом, поскольку при отображении формы тратится время на выборку подформ [4]. Для достижения гибкости внешнего исполнения с минимальными потерями производительности было создано всего две подформы:

- **верхняя часть (top).** Эта часть содержит начало разметки, стартовые теги таблицы, вычисление оглавления разделов первого уровня, если текущий раздел – первого уровня, то вычисление оглавления этого раздела по ключу – имени текущего раздела и т.д. до четвертого уровня. Оглавление разделов отображается на любой странице, оглавление подразделов – в зависимости от того, на каком уровне пользователь находится. Здесь же располагается открывающий тег <td> табличной ячейки, содержащей основное тело документа;

- **нижняя часть (bottom).** Эта часть обычно следует за телом документа, начинаясь с закрывающего тега </td> табличной ячейки, содержащей основное тело документа. Здесь вычисляется информация об ответственном на основе имени пользователя, занесенного в документ при записи документа в поле "автор". Завершается описание нижней части закрываю-

щими табличными тегами и традиционной информацией об авторском праве.

Элементы оформления

Наряду с элементами структуры в главной базе данных были созданы также элементы оформления в виде документов, имеющих имя и набор присоединенных файлов, либо исходный текст и созданных по отдельной форме. Эти элементы могут содержать:

- **присоединенные изображения.** В этих документах хранятся графические изображения (рамки, значки, баннеры, логотипы) (рис.5);

- **присоединенные файлы.** Некоторые, необходимые для статической части, файлы, присоединенные и доступные для скачивания ;

- **код HTML.** Эти документы не содержат присоединенных файлов, в них находится исходный текст HTML, используемый модульно при формировании страниц;

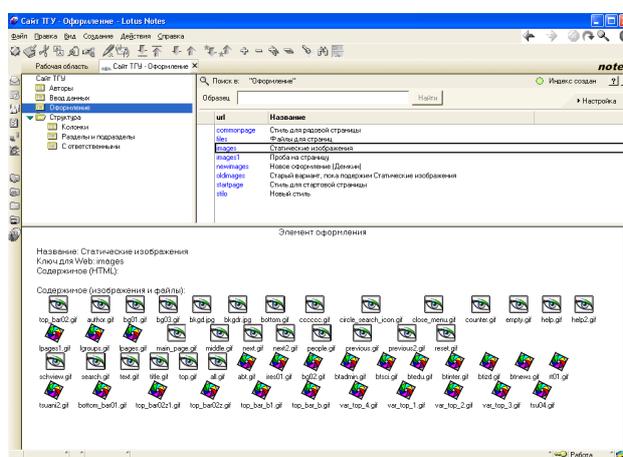


Рис. 5. Форма элемента оформления

- **код JavaScript.** Документы без присоединенных файлов, в них, так же, как в HTML-коде, приводится исходный текст на языке JavaScript (описание переменных и функций и пр.). С точки зрения оптимизации загрузки отображаемых данных в браузер, такие описания можно также помещать в присоединенные файлы с расширением .js – тогда браузер расценивает такой модуль как файловый ресурс и в состоянии его кэшировать;

- **код CSS, XSL.** Так же, как HTML- и JavaScript-документы, эти элементы дизайна содержат описания таблиц каскадных стилей, классов, псевдоклассов. Страницы обращаются к таким элементам посредством тега <link rel=stylesheet type="text/css" href="путь к CSS"> [5].

Виды для поиска и заполнения

Структурные элементы сайта хранятся в главной базе данных. В качестве поисковых ключей в виде используются поля внутренних ссылок. При генерировании страниц осуществляются запросы к этим видам. Генерируемая страница запрашивает по ключу название ссылки и саму ссылку, в зависимости от поля "тип ссылки". Если ссылка внутренняя, то конечная полная ссылка формируется приписыванием пути до сердцевины базы от корня сайта без указания имени сайта и

префикса протокола "http://". Если ссылка внешняя, подставляется значение из поля внешней ссылки.

Список разделов. В этом виде отбираются документы первого уровня (т.е. те, у которых поле уровня пустое) (рис.6).

#	имя	url
1 0	Новости	/WebDesign/TSU/tsu_news.nsf/allnews
1 1	Актуальные рубрики	/webdesign/tsu/core.nsf/structurl/first
1 2	Общие сведения	/WebDesign/TSU/core.nsf/structurl/common
1 3	Абитуриенту	/WebDesign/TSU/core.nsf/structurl/abitur
1 4	Образование	/WebDesign/TSU/core.nsf/structurl/education
1 5	Наука	/WebDesign/TSU/core.nsf/structurl/science
1 6	Международное сотрудничество	/WebDesign/TSU/core.nsf/structurl/intercooperation
1 7	Издательская деятельность	/WebDesign/TSU/core.nsf/structurl/publish
1 8	Культура, спорт и здоровье	/WebDesign/TSU/core.nsf/structurl/culture
1 9	Студенты	/WebDesign/TSU/core.nsf/structurl/student
1 10	Общественные организации	/WebDesign/TSU/core.nsf/structurl/publicbodies
1 11	Интернет-ресурсы	http://www.list.tsu.ru/
1 12	Форумы	http://ic1004.ic.tsu.ru:81/forum/index.html
1 90	Ввод данных	/webdesign/tsu/core.nsf/structurl/enter
2 1	Расписание занятий	/webdesign/tsu/core.nsf/structurl/timetable
2 1	Основные направления научной деятельности	/WebDesign/TSU/core.nsf/structurl/sciactivity
2 1	Специальности и направления	/WebDesign/TSU/core.nsf/structurl/directions
2 1	Интернет-ресурсы	http://www.list.tsu.ru/
2 1	125-летие ТГУ	/webdesign/tsu/core.nsf/structurl/agetsu
2 1	Профком сотрудников	/WebDesign/TSU/core.nsf/structurl/tucstaff
2 1	Приветствие ректора	/webdesign/tsu/core.nsf/structurl/greeting
2 1	Анонсы	/WebDesign/TSU/tsu_news.nsf/allnews
2 1	Факультеты, специальности	/webdesign/tsu/core.nsf/structurl/faculties
2 1	Вестник Томского государственного универси	/WebDesign/TSU/core.nsf/structurl/vestnik
2 1	Центр культуры	/webdesign/tsu/core.nsf/structurl/center
2 1	Общая информация	/WebDesign/TSU/core.nsf/structurl/information
2 1	Ввод данных страницы	/webdesign/tsu/core.nsf/enter

Рис. 6. Список разделов

Вид используется страницами при генерации навигации по разделам сайта. Первая колонка – ключевая, по ней делается запрос. Вторая и третья колонки возвращают соответственно название раздела (имя страницы в структуре) и ссылку на раздел (внутреннюю или внешнюю).

Оглавление раздела. В этом виде отбираются все документы структуры, у которых многозначное поле уровня содержит лишь одно значение, т.е. все элементы второго уровня. Первая, ключевая колонка содержит имя родительского раздела (раздела первого уровня) с тем, чтобы по ключу-имени раздела можно было получить список документов-подразделов запрашиваемого раздела.

Вторая и третья колонки возвращают соответственно имя подразделов и их ссылки (так же – внутренние или внешние) (рис.7).

Абитуриенту	Факультеты, специальности	/webdesign/tsu/core.nsf/structurl/abitur_facult
Абитуриенту	Уровни образования и формы обучения	/WebDesign/TSU/core.nsf/structurl/abitur_edu
Абитуриенту	Правила приема	/WebDesign/TSU/core.nsf/structurl/abitur_adr
Абитуриенту	Вступительные экзамены	/WebDesign/TSU/core.nsf/structurl/abitur_exa
Абитуриенту	Платные образовательные услуги	/WebDesign/TSU/core.nsf/structurl/abitur_ser
Абитуриенту	Отдел по новому набору и доузовской по	/WebDesign/TSU/core.nsf/structurl/abitur_pre
Абитуриенту	Подготовительные курсы и отделения	/WebDesign/TSU/core.nsf/structurl/abitur_coc
Абитуриенту	Эксперт по ЕГЭ	/webdesign/tsu/core.nsf/structurl/abitur_ege
Актуальные рубрики	125-летие ТГУ	/webdesign/tsu/core.nsf/structurl/first_agetsu
Актуальные рубрики	Университет сегодня	/webdesign/tsu/core.nsf/structurl/first_today
Ввод данных	Ввод данных страниц	/webdesign/tsu/core.nsf/enter
Ввод данных	Служба новостей	/WebDesign/TSU/tsu_news.nsf/enternews
Ввод данных	Диссертационные советы	/WebDesign/TSU/dis_council_news.nsf/enter
Ввод данных	Ученые советы	/WebDesign/TSU/sci_council_news.nsf/enter
Издательская деятельн	Вестник Томского государственного унив	/WebDesign/TSU/core.nsf/structurl/publish_vt
Издательская деятельн	Известия вузов. Физика	http://www.almanet.tsu.ru
Издательская деятельн	Газета "Alma Mater"	http://www.almanet.tsu.ru
Издательская деятельн	Редакционно-издательский отдел ТГУ	/WebDesign/TSU/core.nsf/structurl/publish_ic
Интернет-ресурсы	Интернет-ресурсы	http://www.list.tsu.ru/
Интернет-ресурсы	Дискусии и форумы	http://ic1004.ic.tsu.ru:81/forum/
Культура, спорт и здоро	Центр культуры	/webdesign/tsu/core.nsf/structurl/culture_cent

Рис. 7. Оглавление раздела

Оглавление подраздела. Еще один вид, отсекающий только документы структуры третьего уровня, т.е. документы, у которых поле уровня содержит ровно два значения – название корневого раздела и его подраздела.

Ввод данных. Это рабочий вид в головной базе данных, предназначенный для работы ответственных за заполнение частей сайта. Поскольку каждая страница имеет поля типа «читатели», то аутентифицирован-

ные пользователи в одном и том же виде будут видеть только те документы, которые доступны им для редактирования. Такой способ организации рабочих мест для ввода информации позволяет создать только один вид для всех пользователей, распределение доступа происходит автоматически на уровне документов.

Web-интерфейс рабочего места пользователя представляет собой многооконную среду, где в верхнем фрейме предоставляется возможность редактировать документы, а в нижнем открывается библиотека графических ресурсов (рис.8).

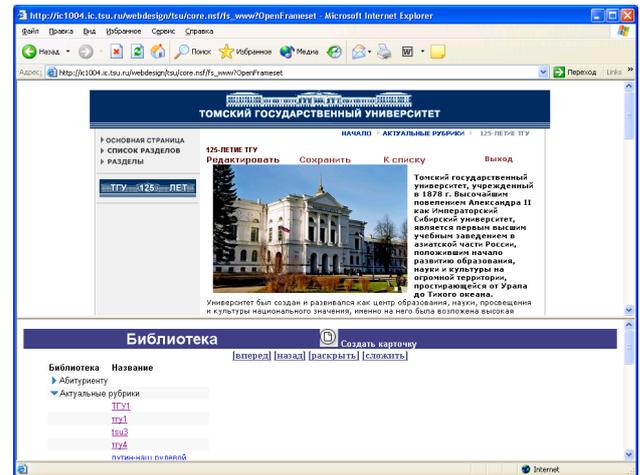


Рис. 8. Web-интерфейс пользователя. Страница сайта в режиме просмотра в верхнем фрейме

Дизайн страниц, единый в пределах всего сайта, определяет форму для ввода, от пользователя же требуется знание основных тегов разметки HTML. Библиотека изображений физически находится в другой базе данных и является общей для всех авторов.

Агенты

Агенты используются при построении сайта в рабочих местах пользователей, заполняющих сайт. Агент запускается по определенному событию, например, при попытке сохранения документа в браузере.

В частности, агенты используются при присоединении файлов к документам – документов Word или файлов изображений (рис.9). В качестве языка был использован язык Lotus Script.

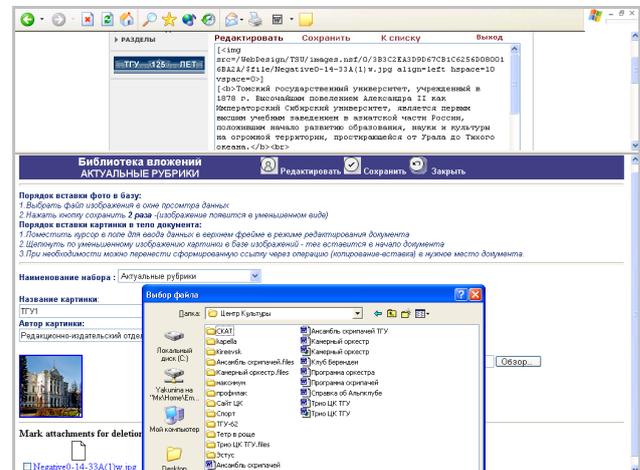


Рис. 9. Карточка графического ресурса

После того, как пользователь создал документ, присоединил к нему файл и попытался его сохранить, агент вычисляет имя присоединенного файла и заносит его в соответствующее поле – имя файла для последующей генерации ссылки на этот присоединенный к документу файл [6].

После того, как документ с присоединенным файлом сохранен, его можно использовать в других документах, создавая в них ссылки на ресурс Интернет (рис. 10).

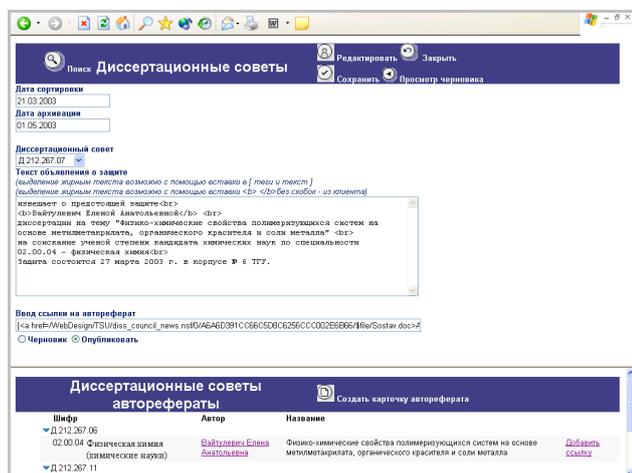


Рис. 10. Web-интерфейс аттестационного отдела. Пример вставки ссылки на автореферат в форму объявления о диссертационном совете

При реализации автоматического вычисления ссылок был использован обработчик события формы «WebQuerySave», который, как видно из названия, активизируется при попытке сохранения, т.е. до фактического сохранения документа.

Суть обработчика состоит в том, чтобы получить список имен присоединенных файлов и создать соответствующее многозначное поле для последующей генерации ссылок. Присоединенный же файл появляется в списке только после сохранения документа. Поэтому авторы столкнулись с проблемой «двойного сохранения» при присоединении файла.

Генерация же ссылок заключалась в склеивании строки (при помощи @-формулы) с тегами со списочным полем имен присоединенных файлов:

```
"[<a href=javascript:putURL(""+@ReplaceSubstring(@Subset(@DbName;-1);"\";"/")+@Text(@DocumentUniqueID)+"/$file/"+images+"")><img src=""@ReplaceSubstring(@Subset(@DbName;-1);"\";"/")+@Text(@DocumentUniqueID)+"/$file/"+images+" width=100 height=100 alt=""@images+"></a>]"
```

Функция putURL была реализована на языке JavaScript:

```
function putURL (URLstring) {
    parent.text.document.forms[0].text.value = "[<img src=""@URLstring+" align=left hspace=15 vspace=0>]" + parent.text.document.forms[0].text.value
}
```

Обработчики событий JavaScript.

В приложениях пользователей – других базах данных сайта (новости ТГУ, новости диссертационных советов ТГУ, ученый совет ТГУ и др.) – созданы спе-

цифичные формы для ввода и поиска информации, содержащие свойственный конкретным подразделениям инструментарий.

При этом для клиентов web (браузеров) разработаны "действия", облегчающие подключение ресурсов интернета к редактируемым страницам. В описании страницы существуют описания функций JavaScript, позволяющие помещать в документ, находящийся в одном фрейме браузера, ссылку на открытый ресурс в другом фрейме браузера. Другими словами, пользователь, видя предварительный просмотр изображения, нажимает на изображения, получая в редактируемом документе ссылку на него (рис. 11).

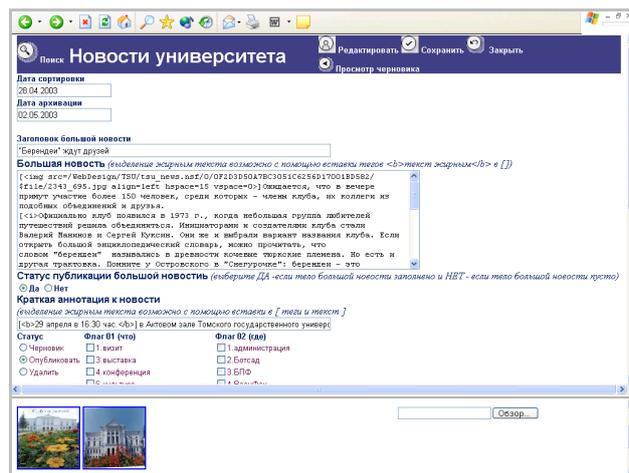


Рис. 11. Web-интерфейс службы новостей. В верхнем окне открыта форма на редактирование

Другим примером использования JavaScript могут послужить вызовы различного рода диалоговых окон.

Определение общей функции показа диалогов.

```
function showMenu(name) {
    hideAllMenu();
    if (name == "AuthorDetails") {
        document.all[name].style.pixelTop = document.body.clientHeight+document.body.scrollTop-200;
        document.all[name].style.pixelLeft = (document.body.clientWidth)/2+60;
        document.all[name].style.visibility = 'visible';
    }
    else
    {
        document.all[name].style.pixelLeft = (document.body.clientWidth-640)/2+130;
        document.all[name].style.visibility = 'visible';
    }
}
```

Вызов диалогового окна «Автор».

```
<a title="послать почту" href=mailto:onMouseOver="showMenu('AuthorDetails'); return false;" onMouseOut="hideMenu('AuthorDetails'); return false;">Иванов И.И.</a>
```

Вызовы списка разделов и оглавления раздела.

```
<a href="" onClick="showMenu('Menu1'); return false;"><img src=/webdesign/tsu/core/nsf/designobjects/images/$file/lgroups.gif width=160 height=17 border=0 alt="Список разделов сайта"></a>
```

```
<a href="/" onClick="showMenu('Menu2'); return false;"><img src=/webdesign/tsu/core.nsf/designobjects/images/$file/lpages.gif width=160 height=17 border=0 alt="Оглавление раздела"></a>
```

РЕАЛИЗАЦИЯ ССЫЛОЧНОСТИ

При построении главной базы сайта, регистрирующей как внутренние, так и внешние ссылки, необходимо было создать механизм для получения навигации всех уровней в каждой базе, являющейся частью одного сайта, избежав дублирования информации и множественных запросов при генерации страниц. Это достигнуто путем использования нескольких механизмов:

- **наследование дизайна.** Главная база (core.nsf) была объявлена шаблоном баз данных. Таким образом, унаследованные другими базами сайта подформы top и bottom приводятся в актуальное состояние сервером Domino по расписанию ежечасно;

- **переменные-указатели на головную базу.** В самой головной базе и в других базах данных были созданы поля, вычисляемые при открытии и содержащие путь к головной базе при запросах навигации. Таким образом, привязка к головной базе была осуществлена введением всего лишь одной константы, это придает сайту мобильность в случаях, когда файлы будут необходимо переместить в другое место на сервере.

- **имя и путь.** Каждая страница, осуществляющая поиск контекстно зависимой навигации, должна передать в запросе в качестве параметра свое имя и путь. Так же, как и в головной базе, в других базах в соответствующих формах для этой цели должны быть созданы поля, значения которых будут использоваться в подформе top. Эти поля могут содержать как константу (если, например, все документы, созданные по конкретной форме, относятся к одному и тому же разделу сайта), так и значения, вводимые пользователем (в этом случае каждый документ, имеющий свой путь, должен быть дополнительно зарегистрирован в структуре – в головной базе).

СТАТИКА, ДИНАМИКА И ВРЕМЯ ЖИЗНИ

Теоретически все множество страниц сайта можно разделить на три группы по степени изменчивости:

1) статические страницы. Такого рода страницы всегда присутствуют на сайте и практически никогда не изменяются;

2) динамические страницы. Информация, представленная на такого рода страницах, актуальна только определенное, фиксированное время;

3) страницы с произвольным временем жизни. Для такого рода страниц задается промежуток времени, в который они будут доступны пользователям сайта.

Естественно, каждый корпоративный сайт, имеющий не только представительскую, но и информационную направленность, содержит и статические, и динамические данные. В техническом плане «устаревшие» страницы можно делать просто недоступными, проверяя, к примеру, дату и время их создания, можно перемещать в архив или удалять из рабочей базы данных сайта. Динамическими являются страницы, ссылки на которые располагаются на первой странице сайта, а также практически все страницы баз данных сайта, имеющих новостную направленность.

Для того чтобы автоматизировать механизм слежения за актуальностью страниц сайта, возможно создание специализированного агента, запускающегося на сервере периодически и проверяющего актуальность всех динамических страниц. Для того чтобы отделить устаревшие естественным образом страницы (например, новости) от страниц, потерявших актуальность и требующих обновления в формах этих страниц, можно создать соответствующие поля (например, «важность» и «периодичность обновления»). Тогда агент, нашедший важные устаревшие документы, использует в них указанное выше поле «авторы» с тем, чтобы напомнить по электронной почте авторам обновить информацию на сайте.

РАБОТА В КЛАСТЕРНОЙ СРЕДЕ

Lotus Domino позволяет строить приложения, в т.ч. и для Web-сайтов, с автоматическим распределением нагрузки между серверами.

Простейший случай, позволяющий работать сайту на нескольких серверах, – это репликация всех баз данных средствами Domino. Для того, чтобы ссылки продолжали вычисляться должным образом, изначально ссылочность необходимо реализовывать относительно корня сервера. При этом накладывается дополнительное ограничение на расположение баз на серверах: они должны располагаться в тех же каталогах данных, либо администраторы серверов должны создать соответствующие символические ссылки [7].

Для того, чтобы сайт, располагающийся на нескольких серверах Domino, имел единый дизайн, необходимо создать реплики главной базы сайта на всех серверах – членах сайта, и на периодической основе реплицировать изменения в этой базе, а также обновлять в других базах унаследованный из этой базы набор структурных элементов.

ЛИТЕРАТУРА

1. Линд Дебби, Керн Стив. Lotus Notes и Domino R5. Энциклопедия пользователя: Пер. с англ./Дебби Линд, Стив Керн. К.:Изд-во «Диасофт», 2000. 656 с.
2. Ионцев Н.Н., Поляков Е.В., Таранченко О.Г. Программирование в Lotus Domino R5: формулы и функции, язык LotusScript, встроенные классы LotusScript и Java. М.: Изд-во «Светотон», 2000. 936 с.
3. Domino R5 Developer's Handbook. IBM RedBook sg245331.
4. Domino R5 Designer Help. IBM Lotus Software.
5. Карандин С., Колосов М., Поляков Е. Краткий справочник по разработке Web-приложений на базе Lotus Domino R5. М.: Изд-во «Светотон», 2000. 331 с.
6. Поляков Е.В. Средства разработки приложений в Lotus Domino R5: Domino Designer. М.: Изд-во «Светотон», 2002. 468 с.
7. Domino R5 Administration Help. IBM Lotus Software.

Статья представлена кафедрой теоретических основ информатики факультета информатики Томского государственного университета, поступила в научную редакцию 11 мая 2003 г.