

**Российская академия наук  
Сибирское отделение  
Институт систем информатики  
им. А. П. Ершова**

**В. С. Рыжов**

**ОБЪЕКТНЫЙ ПОДХОД К СОЗДАНИЮ СТАНДАРТА  
ДОКУМЕНТООБОРОТА В СРЕДЕ ИНТЕРНЕТ**

**Препринт  
109**

**Новосибирск 2004**

Данная работа посвящена обсуждению организации документооборота в среде Интернет. В работе вводится расширенное понятие документа, выбираются наиболее перспективные и надежные форматы документов и технологии доступа к ним, а также рассматриваются варианты взаимодействия документов в сети. Таким образом, в работе предлагается некоторый стандарт на организацию документооборота в среде Интернет.

**Siberian Division of the Russian Academy of Sciences  
A. P. Ershov Institute of Informatics Systems**

**Vladimir S. Ryzhov**

**OBJECT APPROACH TO CREATION OF STANDARD  
FOR DOCUMENT WORKFLOW IN INTERNET ENVIRONMENT**

**Preprint  
109**

**Novosibirsk 2004**

In the work the arrangement of document workflow in the Internet environment is discussed. The author introduces a general concept of document and proposes a mechanism of document workflow using Object Managers, considers existing document formats and technologies of access to the documents and then chooses the most promising and reliable ones for using them as some standard for document workflow in Internet.

## ВВЕДЕНИЕ

Интернет все больше входит в нашу повседневную жизнь. Это уже не просто место, где можно найти что-нибудь интересное, а очень удобный, порой незаменимый помощник во многих областях деятельности. Через Интернет можно заводить новых друзей и находить деловых партнеров, покупать книги и машины, получать образование и просто играть. Трудно перечислить все области деятельности, в которых успешно используется Интернет. Но для того чтобы воспользоваться огромными возможностями всемирной паутины, помимо физического доступа к ней, необходимы механизмы поиска в огромной сети. Здесь и начинаются трудности, с преодолением которых можно будет говорить о доступности знаний, накопленных человечеством.

Насколько разнообразна информация, доступная через Интернет, начиная с космических технологий и заканчивая кулинарными рецептами, настолько разнообразна и велика аудитория пользователей всемирной сети. Неподготовленному пользователю бывает нелегко найти что-то в море неупорядоченной информации. Популярные поисковые системы могут дать только набор документов с известных им серверов, и чем больше ссылок по теме выдает поисковая система, тем больше вероятность того, что в этот список вошли копии одного и того же документа, его устаревшая и новая редакции и т.п. Выбрать действительно актуальные данные довольно сложно. Каждая копия при появлении становится самостоятельной единицей информации, никак не связанной с оригиналом. Кроме того, в список найденных могут попасть случайные документы, например, реклама товаров, в то время как вы искали научную статью.

Но даже если пользователь нашел то, что искал, еще не факт, что он сможет этим воспользоваться. Существует большое количество форматов данных и еще большее количество программного обеспечения для работы с ними. К сожалению, у пользователя не всегда имеется возможность найти программу для работы с данными в специфичном формате. Причем трудно решить, кто тут виноват — тот, у кого есть оригинальная программа, которая хранит данные исключительно в своем формате, или тот, у кого этой программы нет или есть только устаревшая версия.

При углублении в проблему доступа к найденной информации обнаруживается и более серьезная проблема, чем просто отсутствие нужной программы. Практически все программы можно запускать только на определенных платформах, в определенных операционных системах. На этом уровне и данные, и программы для работы с ними находятся в одинаково

зависимом положении. Поскольку эти объекты ведут себя одинаково с точки зрения каталогизации и хранения, далее в статье объединим эти понятия, и будем называть их просто объектами.

Для облегчения работы с объектами у них должны быть: версия, уникальный идентификатор (в настоящее время это просто название документа или программы), и желательно иметь дополнительное описание, позволяющее заранее знать, с чем имеет дело пользователь (например, является ли найденная программа текстовым или графическим редактором). Объект с подобным описанием далее будем называть документом, а само описание — «карточкой» документа.

Следует отметить, что в сети Интернет есть не только «пользователи-читатели», но и «пользователи-писатели». А процесс опубликования документов для неспециалиста труден и непрозрачен. Публикацией документов, созданием web-страниц занимаются только специалисты web-дизайнеры, и тратят они на это немало времени. Несмотря на то что существует стандарт языка HTML, созданного специально для публикации в Интернет, есть большое количество программ для работы с HTML-документами, не поддерживающих этот стандарт полностью. Поэтому львиная доля времени уходит на то, чтобы web-страница выглядела одинаково хотя бы в наиболее распространенных программах просмотра (Netscape Navigator и Internet Explorer).

Эти и другие не столь очевидные проблемы мог бы решить общий стандарт на формат документов, технологию создания и доступа к ним. Единый формат для всех документов, конечно, невозможен, как невозможно в одном формате хранить текст, электронные схемы и музыку, но можно стандартизировать карточки документов и формат для web-страниц. Желательно ввести и логический порядок в хранимые данные, в первую очередь для облегчения поиска нужной информации. Решению этих проблем автор посвятил данную статью, опираясь на свой опыт работы с сетью Интернет как в качестве пользователя, так и в качестве web-разработчика и аналитика.

## **ПОСТАНОВКА ЗАДАЧИ**

Данная работа посвящена обсуждению решения для организации документооборота в среде Интернет. Предлагается изменить логическую структуру сети, выбрать наиболее перспективные и надежные форматы документов и технологии доступа к ним, рассмотреть варианты взаимо-

действия документов в сети. Таким образом, предлагается стандарт на организацию документооборота в среде Интернет. Здесь документ — обобщенное понятие, которое будет рассмотрено ниже.

Для того чтобы предложенный стандарт не остался на бумаге, а получил распространение, необходимо выполнить требования, которые ставит жизнь перед системами, претендующими на всеобщий стандарт.

1. *Простота* — пользователь любого уровня подготовки должен легко находить нужные документы, создавать их и публиковать в сети. Это должно быть отражено в выборе технологии хранения документов.
2. *Четкая структура* — чем прозрачнее логическая структура хранения документов, тем легче провести синхронизацию данных, отследить копии и версии документов, организовать копирование и поиск и уменьшить тем самым общий трафик сети.
3. *Однозначная идентификация документов* — каждый документ должен иметь уникальный идентификатор (предлагается использовать уникальный номер) и информацию о версии документа, что дает возможность автоматического обновления документов-копий до новой версии документа-оригинала и, очевидно, расширит возможности поиска [1].
4. *Наличие карточки документа* — документ должен содержать некоторый набор дополнительной информации, которая будет использоваться как программами-редакторами, так и для упорядочения данных в сети. Обязательным условием при этом является возможность безболезненного расширения количества полей в карточке документа. Такой подход удобен для создания специализированных каталогов, дающих возможность построения и поиска документов по специфичным критериям.
5. *Единый формат данных* — требование относится и к карточкам документов, и, насколько это возможно, к самим документам. Так, в частности, необходимо дать возможность различным производителям программного обеспечения создавать специализированные программы с закрытым исходным кодом, которые эти производители могли бы продавать. При этом в карточке должно быть указано, с каким из стандартных форматов данных работают эти программы. В этой ситуации производители будут продавать не закрытый черный ящик, а только визуальный интерфейс, работающий с известным форматом. Это послужит пресечению создания новых и новых форматов данных.

6. *Единый интерфейс* — обеспечение поддержки программными продуктами на различных платформах и операционных системах. Во избежание наблюдаемых сейчас проблем, связанных с разницей в подходах к созданию документов, необходимо, чтобы все программы поддерживали определенный интерфейс, который давал бы возможность поисковым системам и системам хранения в реальном времени обрабатывать документы и управлять их хранением и распространением.

### *Менеджер объектов*

Надежного функционирования системы масштаба Интернет можно достигнуть только грамотной автоматизацией работы с документами. Решением может стать организация всемирной сети как единой глобальной базы данных с древовидной структурой. Для создания и поддержания структуры хранимых данных необходимы автономные программы — менеджеры объектов. Менеджером объектов будем называть среду хранения информации, способную читать карточки документов и формировать по этим данным глобальную древовидную структуру каталогов, выполняя при этом некоторые служебные функции. Корневым каталогом будет каталог, содержащий все каталоги мира, разбитые по тематике документов. Примером такого разбиения может служить система доменных имен, в которой существуют домены верхнего уровня, разбитые по смысловому содержанию. Создание такой структуры должно упростить процесс поиска необходимых документов, который возможен сегодня только с использованием поисковых систем, непрерывно сканирующих Интернет и создающих свои реляционные базы данных, более или менее актуальные на момент обращения к ним пользователя. Подразумевается, что менеджер объектов отслеживает версию документа, историю его редактирования и копирования между каталогами, что должно помогать пользователю в поиске нужной информации. Схема обращения пользователей к документам такой структуры показана на рисунке.

При работе с глобальной сетью посредством *менеджеров объектов* пользователи обращаются к ближайшему известному им менеджеру, а он, в свою очередь, уже общается с другими менеджерами. Менеджер объектов может обращаться к другим менеджерам как по требованию пользователя, так и в плановом порядке, когда, например, трафик не сильно загружен и возникает необходимость произвести обмен служебной информацией. Менеджер объектов в плановом порядке может синхронизовать версии обслуживаемых им документов – поддерживать всегда самую по-



следнюю, индексировать документы по различным полям карточек для оптимального поиска, выполнять другие действия в зависимости от настроек. Пользователь, войдя в Интернет с известного ему каталога и используя менеджеры объектов, может попасть в любую точку глобальной сети. При этом работа с сетью становится похожей на работу с локальными документами при помощи файловых менеджеров, а этим умением владеет любой пользователь. Если бы число каталогов-разделов глобальной сети было соизмеримо хотя бы с их числом на локальной машине, то вполне можно было бы обойтись и без поисковых систем. Но, конечно же, должны существовать системы автоматизированного поиска, и они будут принципиально отличаться от существующих сегодня в первую очередь тем, что смогут вести поиск по описанию документов, а не только по их содержанию. Это позволит создавать сложные запросы.

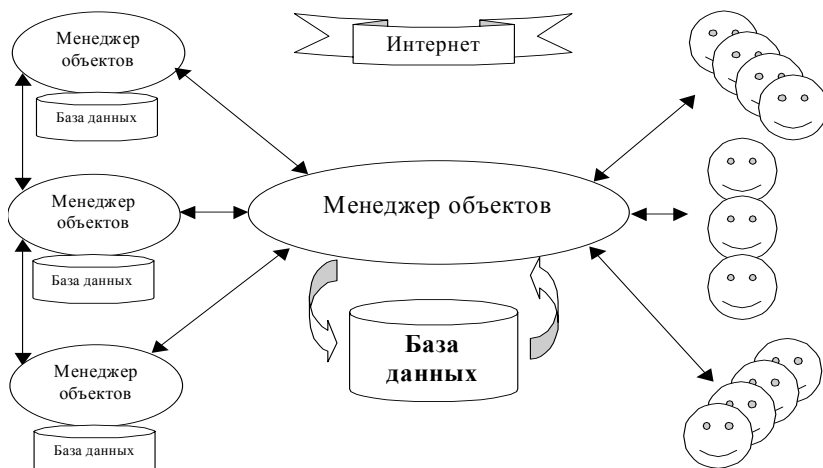


Схема доступа пользователей к документам посредством Менеджера объектов

Используя возможности менеджеров объектов по работе с карточками служебной информации, можно предусмотреть и обратную связь между документом и его копиями. К примеру, можно вести список всех менеджеров, на которых была заведена локальная версия документа. Тогда при появлении новой версии исходного документа содержащий его менеджер документов обращался бы к серверам из этого списка и сообщал бы им об обновлении.

Использование «умных» менеджеров объектов должно помочь пользователю не только в поиске информации и доступе к документам, но и в создании и публикации своих документов. При наличии соответствующих прав доступа к физическим носителям информации пользователь может разместить в сети Интернет созданные им web-страницы или просто архивы документов. В предлагаемой системе достаточно объявить документ «публичным», и он становится доступен всем менеджерам документов, которые по информации о новом документе определяют его логическое размещение в глобальной сети. При этом менеджеры объектов частично формируют содержание карточки документов, т.е. те поля, которые можно заполнить автоматически. Автоматически должно формироваться и уникальное имя документа. Заметим, что это имя может не нести никакой информации о документе как таковом, оно является скорее служебной информацией для менеджеров объектов, а не для отдельных пользователей. Последнее с лихвой компенсируется содержимым стандартизированной карточки документа, где будет храниться информация о версии и первоисточнике документа. Одинаковые для всех документов поля карточки можно использовать при выводе на экран содержимого выбранного пользователем каталога.

## МЕТОДЫ РЕШЕНИЯ

После того как сформулированы требования к стандарту, можно приступить к его более детальной разработке. Для начала надо определиться с выбором технологий, с которыми этот стандарт будет оперировать. Надо понимать, что это очень ответственный момент. Приняв на данном этапе непродуманное решение, можно получить в итоге никому не нужный или нежизнеспособный стандарт. Правильный выбор позволяет значительно сэкономить время и ресурсы на стадии реализации. Очень важен также момент стыковки выбранных технологий. Процесс взаимодействия на стыке технологий не должен быть ресурсоемким как в смысле реализации, так и в смысле использования.

Основным требованием при выборе технологий является возможность оптимального использования уже имеющихся наработок в каждой области. При этом надо стараться не накладывать этим выбором жестких ограничений на результат. Выбирать придется в четырех различных областях.

Сначала нужно выбрать формат, в котором будут храниться документы и карточки к этим документам, не накладывая при этом ограничений на

данные, которые можно хранить в этом формате. На сегодняшний день лидером в этой области является язык XML (Extensible Markup Language). Он получает все большее распространение в области хранения данных, постепенно вытесняя другие форматы. При этом он имеет сильно расширяемую структуру, что позволяет использовать его во многих ситуациях, причем находятся все новые и новые области его применения. Важным является приведение как можно большего числа документов к одному языку написания. Это сделает документы доступными большему числу пользователей и избавит их от необходимости устанавливать новые и новые редакторы и просмотрщики.

Следующая задача, которую необходимо решать, — это уникальная идентификация документов. Для решения задачи глобальной каталогизации и представления документов необходимо иметь уникальный номер для каждой хранимой единицы, чтобы можно было ссылаться на нее. При этом желательно иметь возможность присвоить уникальный идентификатор любому документу на любом локальном компьютере, не обращая к глобальным ресурсам. С этой проблемой может справиться стандартизированная функция, генерирующая некоторым способом уникальные идентификаторы таким образом, чтобы вероятность их повторения равнялась нулю (пример такой функции будет приведен ниже). Такая функциональность неоспоримо расширит возможности системы без наложения дополнительных ограничений.

В качестве уровня представления документов вполне можно использовать обыкновенный HTTP-сервер. Благодаря своей гибкости этот стандарт практически вытеснил большинство своих конкурентов, таких как, например, Gopher. Использование данного протокола позволит применить весь опыт, накопленный в процессе работы с ним, и позволит уменьшить трудности, возникающие при переходе на предлагаемый стандарт логического построения сети Интернет.

Очень важным для будущего стандарта является выбор менеджера объектов. Для составления каталогов следует использовать средства, позволяющие работать с деревьями и ссылками. Самое распространенное из них — протокол LDAP (Lightweight Directory Access Protocol). Он реализован практически для всех платформ. Он достаточно удобен и имеет все необходимое для построения как распределенных, так и локальных каталогов. Кроме того, LDAP имеет гибкий механизм работы с атрибутами хранимых объектов, допускает наличие обязательных и необязательных атрибутов, что позволяет увеличивать количество атрибутов по мере удаления от корневого каталога. В предлагаемой модели это означает воз-

возможность добавления полей к карточке документа. Так, например, формат карточки, описывающей текстовый документ с научной статьей, может сильно отличаться от формата карточки к художественному произведению в том же формате, что является одним из важных требований к создаваемому стандарту.

### **Формат документа**

Сегодня широкое распространение получила технология гипертекста. Активные разработки телекоммуникационных технологий с использованием систем гипертекстовой разметки документов сделали возможным создание интегрированных информационных систем и создание корпоративных ресурсов [3, 5]. Понятие гипертекста было введено В. Бушем еще в 1945 г., и с 60-х годов стали появляться первые приложения, использующие гипертекстовые данные. Всплеск активности вокруг этой технологии начался лишь тогда, когда возникла реальная необходимость в механизме объединения множества информационных ресурсов, а также обеспечения возможности создания и просмотра нелинейного текста [2, 4, 6].

Итак, до недавнего времени основной информационно-структурной единицей электронного обмена данными являлся HTML-документ, который представляет собой обычный текстовый файл, содержащий определенный набор тэгов (ключевые слова, аббревиатуры), заключенных в угловые скобки. В феврале 1998 г. Международный консорциум World Wide Web (w3c) предложил новый язык XML [6].

XML — это метаязык, определяющий другие языки разметки для специфических целей. С его помощью можно описать целый класс объектов данных, передаваемых XML-документами, ориентированными на конкретную предметную область. При этом составитель XML-документа, в отличие от составителя HTML-документа, не ограничен каким-либо фиксированным набором тэгов. Эта возможность является ключевой, она дает возможность по-новому организовать обмен данными в сети Интернет. Это позволяет рассматривать XML как новую технологию в области формирования корпоративных информационных ресурсов, обладающую открытым и эффективным стандартом. Сегодня XML находит все новые и новые области применения. Он может быть использован и в качестве базового стандарта для нового языка описания корпоративных информационных ресурсов, и для создания электронных библиотек на основе баз данных; может быть использован в обычных приложениях для хранения и обработки структурированных данных в едином формате (векторная гра-

фика, текстовая информация). XML-документы, являясь простыми текстами, идеально переносимы между различными операционными системами и компьютерными платформами, что является одним из важных условий всеобщего стандарта. Если в какой-либо предметной области принят единый XML-совместимый язык, то базы данных, написанные с использованием этого языка, можно легко переносить между разработчиками и совместно использовать.

Очевидно, что XML не является панацеей, как и любой другой язык. Хранить все данные только в этом формате не только не эффективно, но иногда и просто невозможно, о чем уже говорилось выше. Важно, что этот язык позволяет унифицировать формат предлагаемых карточек к документам, для этой цели он подходит идеально.

### **Именование документов**

Документ может иметь в среде Интернет несколько представлений: в виде собственно документа, в виде копий документа и в виде ссылок на него. Для того чтобы иметь возможность оперировать с ними, предлагается применить методы объектно-ориентированного программирования, с помощью которых можно построить целостную систему идентификации и регистрации документов в сети Интернет.

Будем рассматривать документ как объект, у которого имеется идентификатор и набор ссылок на него. Первое позволяет отличить его от других объектов, а второе должно служить для определения физического расположения объекта и доступа к нему. Это может понадобиться, например, при составлении карточки документа, когда требуется указать все его первоисточники. Такая информация позволит обновлять копии при изменении оригинального документа, следить за несанкционированным копированием информации и тому подобное.

Обычный пользователь Интернет будет работать с информацией через менеджер объектов и получать информацию, уже обработанную программой-менеджером. Поэтому совсем не обязательно, чтобы пользователю были доступны истинные имена документов. Если с уникальными идентификаторами будут работать только программы, то совершенно не обязательно, чтобы они являлись «именами», которыми документы названы сегодня. Все дополнительные сведения, которые пользователь ранее извлекал из имени документа, теперь хранятся в карточке этого документа. Таким образом, на идентификатор объекта накладывается только условие уникальности. Для оптимальной работы с уникальными идентификатора-

ми необходимо, чтобы они были стандартизованными. Эта проблема может быть решена автоматизацией механизма нумерации для всей сети. Механизм присвоения идентификатора новому документу должен быть надежным и не слишком ресурсоемким.

На сегодня уже существует реально работающий механизм, который можно использовать для решения задачи уникальной идентификации документов. Первой такое решение предложила компания **Open Software Foundation**, которая разработала универсальный уникальный идентификатор — Universally Unique Identifier (UUID). Он представляет собой числовое значение, диапазон которого значительно больше, чем количество объектов, которым его можно присвоить. Определена некоторая хеш-функция, которая вычисляет значения с равномерным законом распределения в этом диапазоне, что приводит к ничтожно малой вероятности совпадения полученных значений.

### **Стандартный HTTP-сервер в качестве сервера хранения документов**

На данный момент протокол HTTP практически полностью занимает нишу по обслуживанию доступа и хранения информационных ресурсов. Протокол Gopher почти не применяется в последнее время, протокол FTP также все больше заменяется протоколом HTTP. Многие потоковые протоколы берут в качестве базы протокол HTTP, используя его как несущий. Протокол HTTP1.1 является расширяемым за счет идеологии заголовков, с его помощью можно реализовать практически любой обмен информацией. По мнению автора, достойной альтернативы протоколу HTTP на сегодня нет. HTTP — это устоявшийся, надежный, проверенный временем протокол, позволяющий решить проблемы хранения и доставки документов.

Для работы с документами на веб-сервере используются специализированные протоколы, расширяющие возможности HTTP, например, такие протоколы, как WebDav.

### **Выбор менеджера объектов**

Поскольку в основе нового стандарта будет лежать использование глобального дерева каталогов, остановимся подробнее на анализе функционирующих в данный момент менеджеров или служб каталогов, а также на достоинствах и недостатках используемой технологии [7, 8].

Рост интереса к службам каталогов вызван тем, что современные информационные системы стремительно увеличиваются в размерах вплоть до масштаба сети Интернет с одновременным расширением спектра пре-

доставляемых услуг. По мере развития системы от локальной сети до сложной распределенной гетерогенной сети на основе различных операционных систем и аппаратных платформ растет и сложность интеграции и управления этими сетями и приложениями.

К недостаткам построения существующих сетей относится то, что в большинстве случаев каждая прикладная задача решается в пределах отдельного приложения или подразделения, при этом используемое программное обеспечение хранит информацию о пользователях, настройках и рабочей среде пользователей в своих собственных конфигурационных файлах. Это приводит к дублированию информации в различных приложениях, усложняет администрирование и увеличивает вероятность ошибок и стоимость владения системой в целом. Ввиду невозможности использования конфигурационной информации одного приложения другими, администратору приходится вносить учетные данные о конкретном пользователе для каждого используемого приложения. Так, например, при смене пароля приходится изменять его во всех приложениях. В то же время, конечным пользователям становится все сложнее находить нужные им информационные ресурсы. Достаточно часто пользователи не подозревают обо всех приложениях, сервисах и ресурсах даже той корпоративной информационной системы, к которой у них имеется доступ, не говоря уже о сети Интернет.

Итак, почему была выбрана именно служба каталогов? Службы каталогов, или, как мы их назвали, менеджеры объектов, уже сейчас являются основным средством построения современных глобальных информационных систем, позволяющим решать как проблемы интеграции информационных ресурсов, так и вопросы управления ресурсами и сервисами сети любого масштаба, вплоть до Интернет. Служба каталога является средством построения стабильной и вместе с тем гибкой сетевой инфраструктуры, предназначенной для обеспечения и контроля взаимодействия пользователей, любых сетевых ресурсов и сервисов.

К основным преимуществам использования службы каталогов в распределенных информационных системах следует отнести:

- создание единого защищенного входа в систему;
- возможность инвентаризации, а в дальнейшем, и интеграции всех информационных ресурсов;
- создание единого способа доступа ко всем приложениям и сервисам и повышение оперативности доступа пользователей на этой основе;

- снижение затрат на администрирование и поддержку информационной системы;
- повышение гибкости создаваемой системы.

Вводимая стандартизация упрощает поиск ресурсов для администраторов и тех пользователей, которые не знают конкретных имен серверов и каталогов. Хорошая служба каталогов делает использование распределенного окружения прозрачным для пользователя. Пользователям не нужно знать расположение файла, приложения или удаленного принтера. Служба каталогов позволяет внести одно изменение, например, в список пользователей, которое затем отразится во всех других ресурсах корпоративной сети.

Служба каталогов предлагает наилучший способ именования, описания и поиска доступных в системе ресурсов, в то же время обеспечивая управление связями между этими ресурсами. Она строится на основе распределенной тиражируемой базы данных, в которой хранятся как логическое описание, так и пути ко всем ресурсам и пользователям в пределах всей корпоративной сети.

Служба каталогов отображает большое количество системных объектов на множество понятных пользователю имен. Основной ее задачей в распределенной сети является описание и поиск сетевых объектов, т.е. пользователей, технических средств, данных, процессов, сервисов, приложений. Эта проблема сложна даже для однородных сетей, так как персонал и оборудование перемещаются, изменяют свои имена и местонахождение. Если говорить о глобальных сетях, то служба каталогов становится намного более сложной из-за необходимости синхронизации различных баз данных каталогов. Более того, при появлении в сети распределенных приложений эта служба должна начать отслеживание всех таких объектов и всех их компонентов.

Служба каталогов распределенной информационной системы может быть построена на основе различных технологий – на службах имен доменов DNS, X.500, LDAP, а также сетевых ОС.

Служба имен доменов, по сути, не является службой каталогов, однако на нее возложены многие функции такой службы. В основном, каталог обеспечивает лишь соответствие имен и IP-адресов. Служба DNS в высшей степени статична, а тиражирование изменений происходит постепенно и медленно. В DNS отсутствует структура для защиты информации и работы с объектами более сложными, чем адреса.

Основные возможности распределенной службы каталогов известны, прежде всего, благодаря Novell Directory Services (NDS). NDS



eDirectory — это полноценная кроссплатформенная служба каталогов, являющаяся основой для различных сервисов менеджера каталогов. Количество приложений, базирующихся на NDS, постоянно увеличивается: автоматизация представления информации, расширенные механизмы безопасности, создание профилей пользователей, использование электронных записных книжек, предоставление автоматизированных систем оповещения, средства автоматизированного создания веб-интерфейсов и механизмы построения виртуальных частных сетей (VPN). eDirectory также является основой для множества собственных продуктов компании Novell, таких как Certificate Server, DirXML™, eGuide, iChain™, Net Publisher и Single Sign-on.

Одним из достоинств NDS является то, что ее каталоги могут объединяться с другими каталогами на «федеративных» началах. Это означает, что пользователи в разных сетях могут иметь различные права доступа к ресурсам своих партнеров. При этом защита информации по-прежнему остается в ведении администраторов отдельных организаций.

Другим примером распределенной службы каталогов является Microsoft Active Directory (AD) — один из наиболее важных компонентов операционной системы Windows 2000. Доменная служба каталогов AD обогащает Windows 2000 рядом новых свойств, которые делают возможным управление глобальными сетями. В числе этих свойств и компонентов — общий каталог Global Catalog, организационные блоки OU, возможность создания расширенных групп, средства репликации каталогов, поддержка протокола LDAP v3. В то же время, механизмы единого управления сетью и единой регистрации пользователей развиты в основном для операционной системы Windows 2000, что затрудняет использование этой службы каталогов для создания гетерогенных сетей.

К сожалению, современные приложения, операционные системы и службы каталогов недостаточно интегрированы между собой. Как AD, так и NDS требуют специальной настройки для работы с определенными программами. Одно из решений проблемы совместимости заключается в использовании промежуточного программного обеспечения для обмена информацией о паролях, идентификаторах и доступе между приложениями. Эту работу выполняет Lightweight Directory Access Protocol (LDAP) — упрощенный протокол доступа к каталогу, клиент/серверный процесс, обеспечивающий стандартный способ хранения, извлечения и создания информации каталогов. В зависимости от используемых служб каталогов можно по-разному структурировать функции аутентификации и авторизации, связав их посредством интерфейса LDAP.

Протокол LDAP постепенно приобретает статус фактического стандарта, регламентирующего доступ пользователей и приложений к информации, которая хранится на серверах каталогов. В настоящее время на рынке присутствует множество серверов LDAP — серверы, пригодные для организации крупных Интернет-узлов, серверы каталогов для корпоративных сетей с интегрированными средствами поддержки LDAP, серверы X.500 с клиентскими приложениями LDAP, а также реляционные СУБД, поверх которых функционируют средства LDAP.

В ряде случаев администраторы из разных подразделений организации не могут прийти к общей схеме наименования ресурсов, или же одно из подразделений при слиянии или приобретении насаждает уже сложившуюся схему. Программный продукт, называемый метакаталогом, позволяет синхронизировать две или более службы каталогов с разными схемами имен. В качестве главного каталога можно пользоваться существующим списком имен, или же метакаталог сам играет эту роль. Определенным образом настроенные метакаталоги могут обмениваться информацией с приложениями, не отвечающими ни одному стандарту каталогов, или с теми, модифицировать которые слишком трудно. Взаимодействие служб каталогов возможно также на основе использования технологий XML. Уже существуют XML-спецификации для каталогов (DSML) и электронных справочников (ECXML).

Естественно, службы каталогов появились не вчера и уже нашли свое применение в глобальной сети.

Службы каталогов сейчас широко используются для поиска объектов и ресурсов распределенных сетей, что позволяет создать эффективные информационно-справочные системы типа "белые/желтые страницы" с числом объектов в каталоге на уровне ста миллионов и более, со временем поиска в несколько секунд. Такие возможности должны стать одной из основных отличительных черт создаваемого стандарта организации данных в сети Интернет.

Особое значение служба каталогов приобретает при создании порталов. Она стирает границы между разрозненными сетями и делает возможным переход к единой управляемой сети. С другой стороны, конфигурация портала, создаваемая службой каталогов, может хранить конфигурации всех пользователей, групп, организационных единиц и организаций, что делает возможным организацию виртуальных сообществ.

На основе службы каталогов возможно создание сетей, в которых информационные ресурсы одной сети будут предоставляться на основе заранее определенной политики пользователям другой сети. Например, часть

информационных ресурсов одного региона может быть доступна для другого региона, или же ресурсы одного ведомства становятся прозрачными для другого.

## **ИСПОЛЬЗУЕМЫЕ ТЕХНОЛОГИИ**

В работе рассматривается выбор технологий для создания стандарта, облегчающего публикацию и доступ к документам в среде Интернет. Если выбранные для всего цикла общения человека с документами технологии достаточно распространены и удовлетворяют основным требованиям к этому процессу, можно предложить этот набор в качестве стандарта, позволяющего упростить жизнедеятельность в среде Интернет. Такого упрощения можно достичь путем уменьшения количества различных технологий, распространенных в данной области применения и не дающих производителю программного обеспечения сконцентрироваться на других, более важных, проблемах. Излишнее разнообразие не дает неподготовленным пользователям возможности ощутить в полной мере все преимущества сети Интернет, ограничивая их лишь теми технологиями, с которыми они знакомы.

### **XML как несущий формат для хранения структурированных данных**

Остановимся более подробно на формате XML, взятом в качестве общего формата документов, на его плюсах и минусах, а также на возможностях его использования при организации структуры сети Интернет.

XML — это набор синтаксических правил, с помощью которых можно описать некоторую структуру, что и должно быть свойственно формату хранения данных. Однако он не справится со своей задачей без программ, которые могут работать с ним. Основное отличие XML состоит в том, что правила формата позволяют определять тэги и использовать их по своему личному усмотрению. Конечно, после этого нелепо ожидать, что документ может быть визуализирован каким-либо браузером, ведь одни и те же тэги в разных документах могут относиться к совершенно разным данным, и к каким именно, не известно никому, кроме автора документа. На самом деле, свобода в рамках XML не так уж велика, и после того как разработчик выбрал имена тэгов, требуется четко соблюдать его немногочисленные формальности.

Для работы с XML уже существует множество доступных технологий, таких как XSL/XSLT, XPath, XDR, XML DOM и так далее, которые по-

зволяют использовать его в реальной жизни. Этими технологиями должен владеть менеджер объектов для работы с карточками документов. Карточки, в зависимости от типа документа, могут иметь различное количество полей, но сохранять при этом обязательные поля. К самим документам, если для них выбран указанный формат, последнее требование не относится.

Выгоды от применения XML в качестве формата карточки очевидны, но насколько он хорош в качестве формата для самого документа? Понятно, что не везде его применение целесообразно. Документы в этом формате не только имеют слишком большой размер, но также требуют больших затрат на расшифровку, чем документы в бинарном представлении. Впрочем, формат XML для документов имеет и достоинства, из-за которых он получает все большее распространение:

- XML позволяет разработчику создать собственную разметку структуры хранимой информации;
- разбор XML хорошо стандартизирован и реализован большим количеством производителей программного обеспечения, что позволяет извлекать информацию из XML-документов практически повсеместно;
- в стандарт XML включена поддержка кодовых страниц Unicode, что упрощает создание многоязычных документов;
- приложения могут использовать XML-парсеры для проверки структуры документов, а при использовании схем — для проверки типов данных, что позволяет значительно упростить разбор строго структурированных документов и тем самым снять с программиста задачу проверки правильности документа;
- XML — это текстовый формат, т.е. читаемый человеком, легко документируемый и иногда более простой в отладке. Оговоримся, правда, что сегодня отладочные средства для ряда XML-технологий пребывают в зачаточном состоянии;
- для работы с XML создано множество средств на самых разных платформах, что делает использование XML более простым по сравнению с бинарными форматами при обмене сложными информационными потоками;
- XML-документы могут использовать значительную часть инфраструктуры, созданной для HTML, включая протокол HTTP и браузеры.

XML может использоваться для представления любой структурированной информации, причем масштабы и сложность этой структуры зна-

чения не имеют. В основном он применяется для хранения служебной информации, как и в нашем случае.

Приведем примеры задач, для которых применим XML.

На базе XML создана технология SOAP, позволяющая осуществлять программные вызовы методов объектов через HTTP в текстовом формате. Говоря коротко, это один из вариантов маршallingа, но не в компактную бинарную форму, а в текстовый формат с большим количеством метайнформации. Это может показаться невероятным, но в условиях наполненного брандмауэрами Интернета это единственный способ вызывать методы объектов.

XML применим также при разработке сложных информационных систем с большим количеством приложений, которые связаны между собой потоками информации, обладающими различной структурой. В этом случае XML-документы выполняют роль универсального формата для обмена информацией между отдельными компонентами большой программы. Сегодня на первый план выходит технология Web Services, использующая XML и SOAP для предоставления программных сервисов в Интернет.

Язык XML позволяет описывать данные произвольного типа, в том числе и специализированную информацию, например: химические, математические, физические формулы, медицинские рецепты, нотные записи и т.д. Это означает, что XML может служить мощным дополнением к HTML для представления в Web "нестандартной" информации. Вряд ли XML полностью заменит собой HTML, но он наверняка сможет упростить жизнь разработчикам и администраторам крупных сайтов.

Из спецификаций XLink и Xpointer родилась вполне приемлемая спецификация XPath (см. ниже), позволяющая создавать краткие и эффективные запросы к структуре документа, ссылаясь как на отдельные элементы документа, так и на группы элементов с учетом вложенности и значений атрибутов.

Использование стилевых таблиц (XSL/XSLT) позволяет упростить изменение структуры и преобразование документов XML или их преобразование в формат HTML.

XML может использоваться в обычных приложениях для хранения и обработки структурированных данных в едином формате. Например, как было сказано выше, в XML можно сериализовать состояние объектов или же хранить документы так, как это делается в новых версиях инструментальных средств фирм Borland и Microsoft.

Большинство из этих решений возможны, но далеко не оптимальны. Реляционные базы данных более продуктивны при работе с большими

объемами информации, и никто не будет заменять их только на основании того, что это возможно сделать. То же относится и к веб-сайтам. Но если обратиться к объектно-ориентированному программированию, то там наоборот, деревья объектов и классов хранятся в реляционном виде, хотя логичнее хранить их в виде древовидной структуры, например, используя XML.

В чем XML неоспоримо превосходит бинарные форматы, так это в том, что информация в нем представлена в текстовом виде. Его можно редактировать в любом текстовом редакторе, XML доступен для понимания человека без дополнительных преобразований, чего нельзя сказать о бинарных форматах, с которыми напрямую работают только программы.

Было бы трудно сказать, зачем XML мог бы понадобиться одному конкретному разработчику. Он подобен языку, который понимает только один человек, но именно на таких языках часто хранится служебная информация для программ. Вся прелесть языка XML в том, что его знают многие, и существуют простые средства для работы с ним, которые можно использовать в готовом виде, а не изобретать самому.

Итак, в XML можно хранить управляющую информацию о ресурсе, а именно: идентификатор, указание категории, класса и другой информации, необходимой для осуществления индексации и управления документом. А сам документ можно хранить рядом с XML-карточкой, и он может быть любого бинарного формата, что, например, очень удобно для хранения запускаемых программ. Эта карточка может включаться в тело документа в качестве заголовка, например, в том случае, если документ хранится полностью в XML-формате.

### **Именованние объектов с помощью UUID**

Проблема именования документов относится не только к всемирной сети, а к хранению данных вообще. Имя для файла данных выбирается уникальное, причем в большинстве случаев автор по этому имени может сказать, какие данные в нем хранятся. С уникальностью имен объектов сталкиваются и программисты, в рамках одного пространства имен имя или цифровой идентификатор переменной, класса, объекта для желаемой работы программы должен быть уникальным. Если вернуться к предлагаемой организации сети, то уникальный идентификатор документа совсем не обязан нести смысловую нагрузку, поэтому можно воспользоваться одним и тем же механизмом именования всех видов объектов всемирной сети Интернет. Если доверить именование документов пользователям,

то неизбежно будут появляться одинаковые идентификаторы, а при добавлении нового документа в сеть нужно будет просмотреть ее всю, чтобы избежать повторов. Трудно сказать, к чему может привести такой подход. В языках программирования (например, C++) эта проблема решается с помощью пространств имен и области действия переменных. Каждый разработчик может давать объектам любые имена и не бояться, что его программа станет обращаться к переменным другой программы. В сети Интернет пространство имен одно, и пользователь, переходя от одного сайта к другому, не должен опасаться, что могут возникнуть конфликты и сбои программ внутри его персонального компьютера. Он должен быть уверен, что перейдет к выбранному менеджеру объектов, а не к его «тёзке».

Итак, система имен объектов должна быть такой, чтобы с какого бы персонального компьютера пользователь ни вышел в Интернет, с каким бы менеджером объектов он ни работал, выполнялось правило уникальности имен объектов. Решается эта проблема переносом ответственности за уникальность имен с разработчика на стандартный алгоритм формирования уникальных идентификаторов.

UUID и есть такое длинное двоичное 128-разрядное число и определённый для него алгоритм вычисления хэш-функции. Значение его, разумеется, не случайное, оно вычисляется с использованием параметров персонального компьютера, таких как идентификационный номер сетевой карты, если она есть, параметры BIOS, значение аппаратного таймера и других. Использование этих величин сводит к нулю вероятность совпадения вычисленных таким образом значений. Это значит, что на каком бы компьютере ни вычислялось значение этой функции, как часто бы это ни делалось, можно быть уверенным, что значение UUID уникально среди всех уже вычисленных и тех, которые еще будут вычислены в мире. Такой подход позволяет решить задачу именования уникальных объектов.

Согласно описанию в книге Крейга Бруксмидта «Inside OLE» [9], алгоритм вычисления UUID базируется на следующих параметрах:

- текущая дата и время;
- последовательность «тиков таймера» и её запоминаемое состояние, чтобы можно было обрабатывать и ситуации обратного передвижения часов;
- простой инкрементальный счетчик, чтобы можно было правильно и достаточно часто обрабатывать запросы на генерацию очередного UUID;
- глобально уникальный IEEE-идентификатор машины, извлекаемый из сетевой карты (реализация алгоритма не требует сетевой

карты; если сетевой карты в машине нет, идентификатор машины может быть синтезирован из непостоянных машинных характеристик и сохранен постоянно).

Точное описание алгоритма вычисления UUID можно найти в [10]. Это часть документации OSF DCE. Глава 10 описывает алгоритм генерации UUID/GUID.

Длина UUID в 128 бит не принципиальна, просто она оказалась подходящей для достижения необходимой вероятности несовпадения значений и удобной в использовании, так как кратна слову процессора, включая самый современный 64-разрядный.

Компания Microsoft воспользовалась этой идеей для именования COM-объектов, но назвала его немного по-другому — GUID, (Globally Unique Identifier, глобально уникальный идентификатор). GUID применяются там, где в модели COM требуется уникально обозначить некое понятие. Обычно обозначаемых ими понятий два — идентификатор класса, называемый CLSID, и идентификатор интерфейса, называемый IID.

Еще раз подчеркнем, что вычисление GUID каждый раз дает разные значения, поэтому этот механизм может использоваться для именования любых объектов по усмотрению программиста или записей в реляционной таблице. Использовать GUID при написании программ для Windows просто, поскольку платформа Win32 имеет функции API UuidCreate и CoCreateGuid, которые вычисляют значение и предоставляют его вызвавшей программе.

Конечно, работать со 128-битным числом, записанным в двоичной форме, неудобно, но так его никто, естественно, не записывает. Для представления GUID существует несколько форм записи. Внутри генерирующей программы существует структура:

```
typedef struct _GUID
{
    unsigned long   Data1;
    unsigned short  Data2;
    unsigned short  Data3;
    unsigned char   Data4[8];
} GUID;
```



т.е. последовательность long-short-short-char-char-char-char-char-char-char.

Если представить эту последовательность в виде шестнадцатеричной записи (каждая шестнадцатеричная цифра заменена символом X), она будет выглядеть так:

```
XXXXXXXX-XXXX-XXXX-XX XX XX XX XX XX XX  
XX.
```

По некоторым структурным причинам последние восемь байтов делятся на два плюс шесть, и последние шесть байтов GUID вычисляются как функция от идентификатора сетевой карты ethernet/tokenring, установленной в машине. Поэтому в символьном виде «настоящий GUID» записывается так:

```
XXXXXXXXXX-XXXX-XXXX-XXXX-XXXXXXXXXXXXXXXXX.
```

Параметры конкретной машины, на которой генерируется GUID, кодируются в последние шесть байт, и в том случае, если на машине есть сетевая карта, это значение будет неизменным. Примеры символьной записи GUID:

```
22F55881-280B-11d0-A8A9-00A0C90C2004  
0c733a60-2a1c-11ce-ade5-00aa0044773d  
B502D1BE-9A57-11d0-8FDE-00C04FD9189D.
```

В записи допускаются символы верхнего и нижнего регистров в произвольном сочетании.

Показанная выше форма записи с разделительными дефисами является стандартной символьной формой, но это не мешает записывать 128-разрядное число в любом другом виде. Его можно представить массивом целых чисел кратной длины, длинным десятичным числом, символами, соответствующими шестнадцатеричным числам. Суть этой величины остается неизменной – другой точно такой же в мире нет. Хотя Win32 и реализует ряд функций API, которые преобразуют GUID из одной формы представления в другую, рекомендуется использовать символьную форму записи. Длина GUID в символьной нотации — 37 символов.

Существует и еще одна символьная форма GUID, в которой подчеркнуто единство всех групп цифр:

```
{B502D1BE-9A57-11d0-8FDE-00C04FD9189D}.
```

В этой форме значение GUID заключено в фигурные скобки. Собственно, только эта форма записи может считаться символьным представлением GUID, поскольку скобки и обозначают формат. Запись без фигурных скобок может быть интерпретирована как просто строковое значение. Но путаницы обычно не возникает, поскольку ситуации употребления этих нотаций различаются.

GUID присваивает объекту его разработчик, а не операционная система, поэтому на всех пользовательских машинах GUID объекта будет один и тот же, и GUID разных объектов у разных разработчиков не пересекутся. Это позволяет теоретически рассматривать случай такого распределённого приложения, часть которого работает на одном континенте, а часть — на другом, и при этом это приложение «ничего не почувствует».

Также нужно знать, что UUID и GUID используются в одинаковом качестве и в COM, и в CORBA, являясь, в сущности, одним и тем же понятием одного назначения, вида и алгоритма вычисления.

### **WebDav как средство размещения документов и доступа к ним**

Для создания возможности публикации документа во всемирной сети и управления ресурсами Интернет чаще всего используется стандарт распределенной разработки и управления версиями Web Distributed Authoring and Versioning (WebDAV), расширяющий протокол HTTP/1.1. WebDAV разработан с целью расширения возможностей протокола HTTP функциями файлового ввода/вывода. RFC 2518 описывает стандарт WebDAV, который позволяет открывать, сохранять, переименовывать, осуществлять поиск, создавать, изменять и удалять файлы на удаленном сервере, совместимом со стандартами WebDAV. Одним из достоинств данной технологии является то, что пользователь может работать с удаленным сервером как с локальным ресурсом. Он может копировать, удалять, сохранять файлы, а также использовать технологию «перетаскивания». Применительно к Novell Enterprise Web Server, клиенты могут работать с удаленными файлами непосредственно из приложений Microsoft Office, рабочего стола Windows 2000 и браузера Microsoft Internet Explorer 5.0.

WebDAV позволяет клиентам:

- работать с ресурсами в каталоге публикации WebDAV на сервере. Это дает, например, возможность пользователям, имеющим соответствующие разрешения, копировать и перемещать файлы в каталоге WebDAV;

- изменять свойства, относящиеся к определенным ресурсам. Например, пользователь может выполнять запись или считывание информации из свойства файла;
- блокировать и разблокировать ресурсы, что дает возможность нескольким пользователям одновременно считывать файл, но только одному пользователю в каждый момент времени изменять его;
- выполнять поиск по содержимому и свойствам файлов из каталога WebDAV.

## **Построение распределенных каталогов на основе LDAP**

Наиболее удачно реализовать идею построения всемирной сети в виде глобального дерева каталогов можно с использованием технологии LDAP, которая уже сегодня распространена достаточно широко.

LDAP появился как упрощенный вариант стандарта X.500, разработанного инженерами Международного Союза Электросвязи (ITU — International Telecommunications Union) в середине 80-х годов. Это был проект по созданию распределенной международной системы каталогов, содержащей данные о телефонных абонентах по всему миру. Эта система должна была располагаться на серверах крупнейших телекоммуникационных компаний.

### *Порядок именованя данных*

Согласно порядку именованя данных, объекты в каталоге упорядочиваются в логическое «дерево»: корень дерева не имеет имени, а вершины характеризуются парами «атрибут—значение». Полное наименование вершины получается путем составления всех таких пар на пути от корня дерева к самой вершине. К примеру, человек может иметь следующий уникальный идентификатор: `sn=Иванов Иван Иванович, ou=Вычислительный Центр, o=ИСИ СО РАН, l=Новосибирск, c=ru`. Здесь `sn`, `o`, `ou`, `l`, `o` и `c` — раз и навсегда зафиксированные названия атрибутов: `c` (`country`) — название страны, `o` (`organization`) — название организации и т.д.

### *Порядок представления данных*

Каждая вершина дерева, помимо уникального названия, содержит информацию о соответствующем реальном объекте, представленную в виде набора пар в том же формате «атрибут—значение». Так, с указанной выше вершиной могут быть связаны пары `"mail: privalov@mail.ru"`, `"phone: +7-123-45678"` и т.д.

## *Клиентский протокол DAP*

Клиентский протокол DAP (Directory Access Protocol) описывает порядок отправки запросов от клиентской программы к каталогу и формат ответа.

### *Протокол для создания "теневого" копии DISP (Directory Information Shadowing Protocol)*

Для повышения эффективности каталогов иногда имеет смысл создать резервную копию, доступную только для чтения (т.е. "тенью"), поблизости от потребителей информации. При помощи DISP один каталог может передать данные другому каталогу для использования в качестве теневой копии.

Протокол для передачи запросов DSP (Directory System Protocol). При помощи протокола DSP один каталог может запросить у другого информацию, которая отсутствует у него самого. Таким образом, клиент, связываясь всегда с одним и тем же каталогом, сможет получать данные из всей системы каталогов.

Построение глобального каталога с использованием стандарта X.500 практически нереализуемо из-за сложности и ненадежности предложенного стандарта. Построение каталогов предприятия, хранение электронных адресов на его основе тоже оказались сложными. Кроме того, создателями стандарта не была предусмотрена возможность запускать клиент DAP на персональных компьютерах и небольших рабочих станциях под управлением ОС Unix.

Решение этих проблем было предложено группой из Мичиганского Университета под руководством Тима Хауса. В 1993 году Хаус с коллегами опубликовали текст RFC 1487 с описанием LDAP (Lightweight Directory Access Protocol) — протокола, работающего непосредственно с TCP/IP и достаточно «легковесного», чтобы хорошо работать на обычных персональных компьютерах.

Для преодоления проблемы сложности и ресурсоемкости серверов X.500 группа Хауса создала простой сервер slapd, который хранил данные непосредственно на локальном жестком диске. Таким образом, каталоги LDAP, сохраняя логические стандарты X.500 (организация объектов в логическое дерево, хранение данных в виде «атрибут-значение», и т.д.), стали независимыми от X.500 на уровне протоколов, полностью перейдя на стандарты эпохи Интернет.

Современные серверы LDAP обеспечивают минимальное время чтения данных, при этом изменение данных происходит гораздо медленнее. Эта особенность привела к тому, что упомянутая система каталогов более эффективно используется для организации справочных систем, где информация гораздо чаще считывается, чем изменяется.

Сегодня нельзя точно определить лидера на рынке каталогов LDAP. Серверы каталогов разрабатываются многими крупными компаниями. Вдобавок внедрение корпоративной системы каталогов остается проектом, требующим активного вовлечения специалистов (обычно консультантов, приглашаемых со стороны); для компаний-производителей серверов LDAP консалтинг становится немаловажной статьёй дохода.

Имеющиеся серверы каталогов можно классифицировать по ряду параметров.

- Связь с основной сетевой операционной системой. В некоторых случаях NOS-каталоги (Network Operating System) вообще не подходят (так, Internet-провайдеру нет смысла заводить учетную запись Windows 2000 для каждого абонента). В других случаях использование NOS-каталога возможно, но не всегда разумно. Например, иногда профили пользователей Internet складываются отдельно от имеющегося NOS-каталога из соображений распределения риска.
- Совместимость с X.500.
- Поддержка создания теневых копий только для чтения (каталоги X.500 автоматически поддерживают создание теневых копий по протоколу DISP).
- Предложение мультимастерной репликации. Мультимастерная репликация означает существование нескольких синхронизированных каталогов, одновременно доступных для записи. Надо заметить, что мультимастерная репликация потенциально опасна: в каталоги LDAP не встроена система контроля транзакций, так что информация об объекте может изменяться одновременно несколькими, возможно, противоречащими друг другу способами (например, в одном каталоге под данной вершиной дерева создается еще одна вершина, а в другом первая вершина удаляется). Тем не менее, мультимастерная репликация бывает полезна, если требуется по-настоящему распределенное администрирование данных каталога.

В каталогах LDAP может храниться любая информация, например, адреса, для которых изначально создавалась структура каталогов; данные

пользователей локальной сети, которые могут применяться для аутентификации. Таким образом, LDAP вполне можно назвать базой данных, преимуществом которой является возможность простого объединения нескольких таких баз данных, что при работе с реляционными данными является весьма сложной задачей. На сегодня реляционные базы данных являются самыми распространенными, и существуют серверы каталогов, которые являются надстройками над реляционными базами данных, например, Oracle Internet Directory и IBM SecureWay. Они более громоздки и менее эффективны, но упрощают работу тем организациям, у которых уже установлены соответствующие базы данных.

Каталоги таких фирм, как Siemens, IBM и Oracle, чаще всего внедряются в рамках более крупных проектов с участием консультантов соответствующих фирм или их партнеров. Только Critical Path предлагает свой продукт вне какого-либо интегрированного набора решений. Свободно распространяемый сервер OpenLDAP основан на slapd Тима Хауса и идеален для обучения технике LDAP; к сожалению, при работе с большими объемами данных OpenLDAP часто бывает нестабилен. В совокупности с общим недоверием ИТ-менеджеров к продуктам категории «open source» это объясняет относительно малое проникновение OpenLDAP в коммерческие организации (в университетах он довольно широко распространен).

Для синхронизации данных между разными каталогами можно использовать механизм метакаталогов (meta-directory). Сам термин «метакаталог» был введен в обиход в 1997 г. аналитиками из Burton Group; в том же году появился первый коммерческий продукт, ZoomIt MetaDirectory (компания ZoomIt была вскоре после этого поглощена корпорацией Microsoft).

При синхронизации данных необходимо произвести создание и удаление документов из соответствующих каталогов, а также синхронизацию значений атрибутов, характеризующих эти объекты. При создании документа в каталоге-собственнике метакаталог создаст ссылки и копии документа на всех подключенных системах, в соответствии с конфигурацией. Для каждого поля карточки документа имеется «авторитетный» источник, причем такие источники могут быть разными для разных атрибутов. Метакаталог берет данные из авторитетного источника и копирует их во все остальные подключенные системы.

Представленные сегодня на рынке метакаталоги довольно сильно различаются по функциональности и архитектуре и не являются взаимно заменимыми.

У данных продуктов имеются следующие отличительные признаки.

- Наличие отдельного метапредставления (MetaView). Метапредставлением называют выделенный каталог, в котором собираются данные из всех подсоединяемых (вторичных) каталогов. У метакаталогов, использующих метапредставление, синхронизация может происходить только между вторичными каталогами и метапредставлением. Напротив, метакаталоги, не использующие метапредставления, обычно позволяют копировать данные откуда угодно и куда угодно (например, из каталога в каталог, из каталога во временный файл или из временного файла в каталог.) Метапредставление часто используется как основной каталог организации.
- Возможность работы с различными типами каталогов. Модули, которые осуществляют поддержку работы с определенным типом каталогов, называются коннекторами. Если специальные коннекторы отсутствуют, обычно предусматривается доступ через отдельно написанные программы (обычно на Perl, JavaScript или VBScript); таким образом, любые данные, доступные соответствующей программе, становятся доступны метакаталогу.

Немаловажным фактором при работе с каталогами LDAP является отлаженность, надежность технологии. Существует большое число продуктов, связанных с LDAP. Производители многопользовательского программного обеспечения встраивают в свои продукты поддержку хранения профилей пользователей в каталоге LDAP. Популярность этих каталогов привела к широкому распространению простых клиентов LDAP (такой клиент, например, встроен в Microsoft Outlook и Outlook Express). Тем не менее, можно выделить ряд продуктов, имеющих более непосредственное отношение к внедрению и администрированию каталогов.

### *Виртуальные каталоги и прокси-серверы LDAP*

Рассмотрим виртуальные каталоги и прокси-серверы LDAP, такие как iPlanet Directory Access Router (iDAR) и MaXware Virtual Directory (MVD). Из этих продуктов наиболее мощным набором функций, имеющих принципиальное значение при организации глобальной сети, обладает MVD: он принимает запрос клиента по LDAP (на чтение или на запись данных) и переводит его либо в другой запрос LDAP к собственно серверу, либо в запрос к реляционной базе данных, либо в запрос информации, реализованный в виде отдельной программы. Таким образом, MVD позволяет добиться функциональности каталога LDAP без установки сервера LDAP: данные могут храниться либо в реляционной базе данных, либо вообще в

файле. MVD может также связываться с несколькими источниками данных с серверной стороны, позволяя распределять данные по нескольким каталогам, либо балансируя запросы клиентов при высокой загрузке путем перенаправления их к нескольким разным копиям данных. iDAR специализируется исключительно на балансировке запросов LDAP между несколькими копиями.

На сегодня рынок каталогов активно развивается. Служба каталогов на основе LDAP, специализированная для работы в среде Интернет, сможет решить проблемы упорядочения, синхронизации, оптимизации поиска данных. Уже существуют реализации LDAP для разных платформ, а использование этой технологии в совокупности с XML, UUID, WebDAV позволит создать общий стандарт хранения данных и работы с ними для решения вопросов документооборота в глобальной сети. Автоматизация функционирования сети Интернет позволит максимально упростить работу с ним простых пользователей.

## ЗАКЛЮЧЕНИЕ

В заключение можно сделать вывод о том, что на данный момент самыми подходящими технологиями, которые могут претендовать на участие в комплексном стандарте по управлению и представлению документов в среде Интернет, могут быть выбраны:

- 1) XML в качестве формата данных (в общем случае, формата карточки к данным, позволяющей формализовать работу с этим документом и его индексацию);
- 2) UUID в качестве алгоритма по созданию действительно уникального идентификатора документа;
- 3) WebDAV через HTTP в качестве стандарта по работе с документом и его публикации;
- 4) LDAP в качестве протокола каталогизации, позволяющего создавать распределенные каталоги и осуществлять быстрый поиск по ним.

Используя эти технологии в комплексе, можно получить жизнеспособный стандарт (особо принимая во внимание, что по отдельности каждая из этих технологий фактически уже занимает лидирующее положение в своей области), описывающий весь цикл работы с различными ресурсами, имеющими хождение в среде Интернет. Выбранные стандарты покрывают весь цикл по работе с различными ресурсами в среде Интернет, некоторые



из них даже немного пересекаются по функциональности. Так, например, WebDAV содержит функции локального поиска, которые также имеются в LDAP. При более тесной интеграции этих протоколов их можно будет немного упростить, убрав дублирующиеся функции и делегировав их той или иной части. Скажем, функции поиска следует, конечно, отнести на LDAP. WebDAV и LDAP вместе образуют менеджер объектов, с которым будет работать пользователь. Использование единого стандарта позволит упростить процесс работы с документами в среде Интернет и позволит избавиться от дублирования функционалов в сравнении со случаем раздельного использования технологий.

### СПИСОК ЛИТЕРАТУРЫ

1. **Марчук А.Г., Осипов А.Е.** К вопросу об идентификации электронных документов и коллекций // Программирование. — 2000. — № 3 — С. 53–62.
2. **Браун М., Ханикатт Д.** HTML 3.2 в подлиннике. — СПб., 1998. — 1040 с.
3. **Кирмайер М.** Мультимедиа. — СПб., 1994. — 192 с.
4. **Питц-Моулитис Н., Кирк Ч.** XML. — СПб., 2000. — 736 с.
5. **Полани М.** Неявное знание. — М.: Прогресс, 1984. — 141 с.
6. **Нау D.** XML: What is it, anyway? // Intelligent Enterprise. — 1999. — Vol. 2, № 11. — P. 12-13.
7. **Чакон М.** Как создать единый каталог // Журнал сетевых решений/LAN. — 2001. — № 5. — С. 32–37.
8. **Чулков А.** LDAP в середине пути // Журнал сетевых решений/LAN. — 2001. — № 5. — С. 38–43.
9. **Brockschmidt K.** Inside OLE. — Microsoft Press, 1995. — 1194 p.
10. **Miller S.** DEC/HP Network Computing Architecture Remote Procedure Call RunTime Extensions Specification Version OSF TX1.0.11. — Open Software Foundation, Cambridge, MA, 1992.
11. **Dublin** Core Metadata Element Set, Version 1.1: Reference Description. — available at <http://dublincore.org/documents/dces/>.
12. **The Dublin** Core Metadata Element Set. Draft Standard. Modified September 30, 2000: ISO TC 46/SC 4, N 515. — 14 p.

**В. С. Рыжов**

**ОБЪЕКТНЫЙ ПОДХОД К СОЗДАНИЮ СТАНДАРТА  
ДОКУМЕНТООБОРОТА В СРЕДЕ ИНТЕРНЕТ**

**Препринт  
109**

Рукопись поступила в редакцию 27.11.03  
Редактор З. В. Скок

---

Подписано в печать 15.03.04  
Формат бумаги 60 × 84 1/16  
Тираж 50 экз.

Объем 1.9 уч.-издл., 2.1 п.л.

---

ЗАО РИЦ «Прайс-курьер»  
630090, г. Новосибирск, пр. Акад. Лаврентьева, 6, тел. (383-2) 34-22-02