



ГОСУДАРСТВЕННЫЙ НАУЧНЫЙ ЦЕНТР
РОССИЙСКОЙ ФЕДЕРАЦИИ
ИНСТИТУТ ФИЗИКИ ВЫСОКИХ ЭНЕРГИЙ

2007–15
На правах рукописи

Сухов Сергей Александрович

**РАЗРАБОТКА СИСТЕМЫ МОНИТОРИНГА
ОПЕРАТИВНЫХ ДАННЫХ
НА ОСНОВЕ WEB-ТЕХНОЛОГИЙ**

05.13.11 – математическое и программное обеспечение вычислительных машин,
комплексов и компьютерных сетей

А в т о р е ф е р а т
диссертации на соискание ученой степени
кандидата физико-математических наук

Протвино 2007

Работа выполнена в Институте физики высоких энергий (г. Протвино).

Научный руководитель – кандидат физико-математических наук, доцент Гусев Виктор Владимирович (ИФВЭ, г. Протвино).

Научный консультант – кандидат физико-математических наук Ухов Владимир Иосифович («Систел», г. Протвино).

Официальные оппоненты – доктор физико-математических наук, профессор Сытин Александр Николаевич (ИФВЭ, г. Протвино), кандидат физико-математических наук, старший научный сотрудник Трошиев Юрий Витальевич (МГУ, г. Москва).

Ведущая организация – Лаборатория информационных технологий Объединенного института ядерных исследований (г. Дубна).

Защита диссертации состоится “ ____ ” _____ 2007 г. в ____ часов на заседании диссертационного совета К 201.004.01 при Институте физики высоких энергий по адресу: 142281, Протвино Московской области.

С диссертацией можно ознакомиться в библиотеке ИФВЭ.

Автореферат разослан “ ____ ” _____ 2007 г.

Ученый секретарь
диссертационного совета К 201.004.01

В.Н. Ларин

ОБЩАЯ ХАРАКТЕРИСТИКА РАБОТЫ

Технологии Интернет вторгаются во все сферы деятельности человека. Не обошли они своим вниманием и производство. Сетевые технологии и удаленный обмен данными играют важную роль во многих промышленных отраслях и, в частности, в энергетике. Данная работа посвящена применению WEB-технологий в области SCADA для мониторинга оперативных данных. Приложения SCADA (Supervisory Control and Data Acquisition System) используются для удаленных измерений, мониторинга критических данных и операций, контроля и защиты, для поддержания надежности и стабильности промышленных систем. Усложнение систем контроля, рост потребности в своевременной оперативной информации и числа клиентов, желающих эту информацию получать, приводят к изменению структур информационных систем, к дополнению, а иногда и замене традиционных технологий.

Разработка описываемых в данной работе приложений была начата в 2000 году в рамках проекта системы мониторинга ускорительно-накопительного комплекса ИФВЭ. Система WEB-мониторинга использовалась для мониторинга технологических данных ускорителя У-70, а также оперативных данных с экспериментальных физических установок ИФВЭ.

В дальнейшем разработанное программное обеспечение получило применение в области энергетики. Описываемые технологии мониторинга данных через Интернет обеспечивают диспетчерам энергосистем возможность удаленного просмотра оперативных данных. Ряд заинтересованных лиц, в том числе дирекция энергопредприятий, может получать оперативную производственную информацию в различных точках сети без установки специальных клиентских программ.

Цель работы. Целью данной работы является разработка программного обеспечения для отображения в Интернет оперативной диспетчерской информации, а также архивных данных.

В разработанное программное обеспечение входят набор графических JAVA-апплетов, HTML-страниц и функций JavaScript, а также ряд подпрограмм, встроенных в специализированный WEB-сервер "Систел".

Специализированный WEB-сервер обеспечивает доставку клиентам событийно-изменяемых данных, а также обеспечивает ряд стандартных функций WEB-сервера. JAVA-апплеты обеспечивают прием и отображение данных, а также навигацию в информационном пространстве.

Актуальность работы. SCADA-приложения применяются во многих областях промышленности. Часто они являются ключевыми звеньями технологии. Допуск программных продуктов, разработанных иностранными фирмами, в область управления стратегическими областями производства, в частности энергетикой, крайне нежелателен. Наверное, многим памятно недавние события в Ираке, когда через спутниковую связь была отключена система противовоздушной обороны этой страны. Делать страну заложником иностранных технологий очень опасно.

Нельзя сбрасывать со счета и экономические аспекты иностранных внедрений. Все SCADA иностранных производителей стоят крайне дорого (сотни тысяч долларов), и кроме того, они являются закрытыми. Поэтому, даже заплатив за них требуемые деньги, заказчик будет вынужден в дальнейшем платить за обслуживание, настройку и модификацию системы. Внедрение иностранных программных продуктов потребует закупки большого количества дополнительного дорогостоящего оборудования, так как иностранные комплексы часто не могут работать с российским оборудованием.

В связи со всем вышесказанным, следует развивать и всячески поощрять национальные разработки SCADA.

Благодаря наличию большого количества WEB-программистов, открытости WEB-технологии и простоте создания пользовательских интерфейсов на ее базе, применение WEB-технологии позволяет существенно удешевить разработку и поддержку подсистем SCADA для удаленного просмотра данных.

Научная новизна:

1. Впервые на базе XML-стандарта разработан специализированный язык для описания мнемосхем, отображающих оперативные данные.

Для отображения событийно-изменяемых оперативных данных на мнемосхемах и отображения самих мнемосхем был разработан специальный язык описания мнемосхем на основе стандарта XML. Файл описания мнемосхемы содержит полную информацию о внешнем виде мнемосхемы и способах реакции на различные события, случающиеся в результате прихода тех или иных оперативных данных или интерактивных действий пользователя. Благодаря единому описанию мнемосхем возникает принципиальная возможность использования мнемосхем разными SCADA.

2. Впервые для подобных систем предложено использовать механизм контроля качества доставки оперативных данных.

Когда речь идет о просмотре данных через глобальную сеть Интернет, предполагается, что клиент может находиться в произвольной точке мира. Глобальная сеть неоднородна по качеству, надежности и скорости передачи данных, поэтому в такой ситуации не может осуществляться передача данных за гарантированное

время. Однако на практике ситуация не столь плоха, тем более, что сетевые информационные коммуникации быстро развиваются, скорость и качество передачи информации растут. При мониторинге оперативных данных важно обеспечить клиенту уверенность в том, что он получил информацию без потерь и за разумный, наперед заданный промежуток времени. Если качество передачи информации в какой-то момент времени упало, клиент должен быть об этом проинформирован. Для этого должен осуществляться непрерывный контроль за временем прохождения данных от точки расположения WEB-сервера до точки расположения клиента. В данной работе предложен механизм такого контроля. Нельзя гарантировать доставку данных за определенный промежуток времени, однако можно контролировать время доставки и сохранность данных, что само по себе крайне важно.

На основе опыта эксплуатации системы мониторинга оперативных данных в реальных промышленных системах в энергетике показана возможность передачи данных за приемлемые промежутки времени.

3. Решена проблема прохождения оперативных данных через Proxy-сервера и Firewall без специальных разрешений и выполнения дополнительных настроек сети.

Рассмотрен ряд проблем, возникающих при передаче оперативных данных через Интернет, в частности проблема прохождения пакетов с данными через прокси-серверы. Отсутствие дополнительных настроек сети для передачи оперативных данных устраняет административные проблемы при внедрении системы, что существенно облегчает само внедрение.

4. Предложены интересные с точки зрения философии WEB-технологий решения для сохранения клиентской информации на стороне сервера в виде конфигурационных файлов.

На основе данной технологии были реализованы механизмы авторизации пользователей системы, механизмы, обеспечивающие индивидуальный доступ пользователей к ресурсам, а также механизмы генерации диспетчерских отчетов. Благодаря встроенному в специализированный WEB-сервер механизму, заполненная пользователем в HTML-формах информация может автоматически сохраняться в полях соответствующих конфигурационных файлов.

Практическая ценность работы. После завершения периода опытной эксплуатации система мониторинга и отображения оперативных данных через Интернет была успешно сдана в промышленную эксплуатацию как часть SCADA “Систел” на предприятии ЦУС (Центр Управления Сетями) “Белгородэнерго”. Ранее система мониторинга была успешно опробована на предприятии “Тамбовэнерго”. Кроме того, в настоящий момент времени система находится в опыт-

ной эксплуатации на предприятии “Московские Высоковольтные Кабельные Сети” и Череповецкой ГРЭС.

ЦУС “Белгородэнерго” является пилотным проектом для SCADA “Систел”. После завершения периода опытной эксплуатации в ЦУС “Белгородэнерго” SCADA “Систел” и, соответственно, система WEB-мониторинга оперативных данных будут ставиться в областных Российских ЦУС, уже сейчас есть договоренности с Ростовским, Воронежским и Саратовским ЦУС.

Автор защищает:

- ◆ Разработку программного обеспечения для отображения в Интернет оперативной диспетчерской информации.
- ◆ Разработку на базе XML специализированного языка для описания мнемосхем, отображающих оперативные данные.
- ◆ Механизм контроля качества доставки оперативных данных через Интернет.
- ◆ Решение проблемы прохождения оперативных данных через Proxy-сервера и Firewall без специальных настроек администраторов сетей.

Апробация работы. Основные результаты диссертационной работы докладывались на научном семинаре отдела математики и вычислительной техники ИФВЭ, на XVIII Конференции по ускорителям заряженных частиц RUPAC-2002, на Международном симпозиуме по проблемам модульных систем и сетей IC-SNET' 2001, на Втором научно-техническом семинаре по современным системам контроля и управления электрических станций и подстанций (АСУ ТП) на базе микропроцессорной техники, проходившем в 2001г. в Москве.

В основу диссертации положено 6 публикаций, две из которых опубликованы в журналах, входящих в перечень периодических научных и научно-технических изданий, рекомендованных ВАК.

Структура диссертации. Диссертация состоит из введения, трех глав, заключения и списка литературы, содержит 13 рисунков. Объем диссертации составляет 89 страниц.

Содержание работы

Во введении формулируется цель работы, обосновывается ее актуальность, рассказывается о практическом применении работы и ее научной значимости.

В первой главе рассматриваются вопросы, связанные с передачей оперативных данных через Интернет. Рассмотрены подходы к удаленному мониторингу

данных, реализованные ведущими разработчиками SCADA, такими как Intellution, WonderWare, AdAstra и др. Предложено решение задачи прохождения оперативных данных через прокси-сервера без специальных настроек сети. Рассмотрена схема передачи оперативных данных через Интернет, а также предложен механизм контроля качества доставки оперативных данных.

Для того чтобы сделать систему более открытой и не лишиться мощных средств отображения информации в виде HTML, был реализован апплет “диспетчер”, позволяющий привязывать событийно изменяемые данные к элементам HTML-страницы и функциям JavaScript.

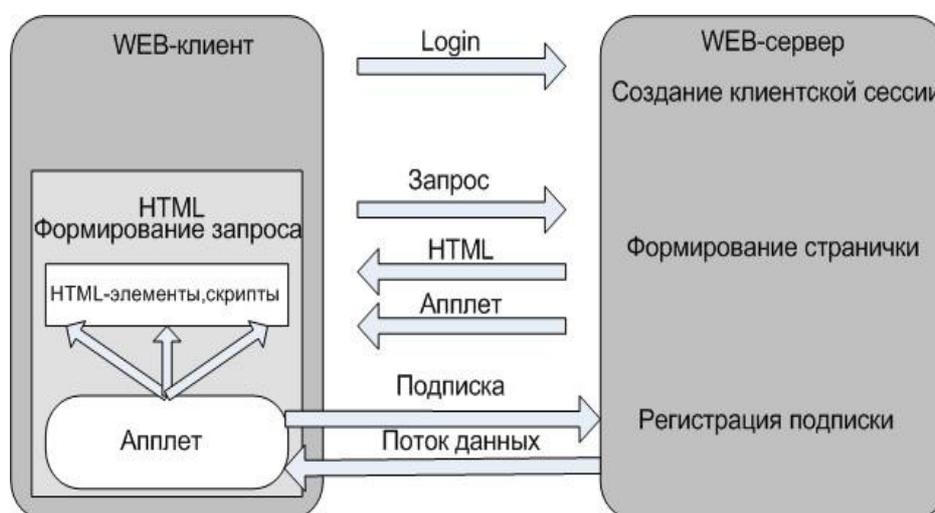


Рис. 1.

Классы апплета **диспетчер** присутствуют в каждом апплете, получающем событийно-изменяемые данные. Апплет-диспетчер невидим для клиента. Задачей апплета является подписка на данные, получение данных, обработка приходящих данных и выполнение операций, связанных с приходом тех или иных данных. Апплет-диспетчер должен восстанавливать соединение с WEB-сервером в случае обрыва связи, а также контролировать время доставки данных. При задержке прихода данных дольше заданного периода времени апплет должен уведомлять клиента о плохом качестве связи. На какие данные должен подписываться апплет и какие действия он должен выполнять по приходу соответствующих данных, апплет узнает из списка параметров.

В значении параметра запаковывается сигнатура данных, на которые подписывается апплет, и информация о том, какой элемент HTML-странички апплет должен изменить, либо какую функцию JavaScript он должен вызвать при приеме данных с соответствующей сигнатурой. Вызываемая функция JavaScript должна иметь два параметра – сигнатуру данных и пришедшее значение.

Выполнение работы апплета происходит в два потока. Первый поток занимается приемом и обработкой данных, в случае обрыва соединения он завершается. Второй поток отслеживает жизнедеятельность первого потока. В случае преждевременной остановки первого потока он производит его перезапуск.

Контроль за качеством доставки может быть активным и пассивным. Пассивный контроль фиксирует сбои сети и создает архив сбоев и временной загрузки различных участков сети. Активный контроль основан на механизме контрольных запросов (ping). Для мониторинга оперативных данных используется активный контроль. Идея заключается в том, чтобы при получении событийно-изменяемых данных встроить механизм контроля качества связи в механизм доставки данных. Задается контрольный промежуток времени, в течение которого данные должны быть получены. Этот промежуток времени может зависеть от поставленной задачи.

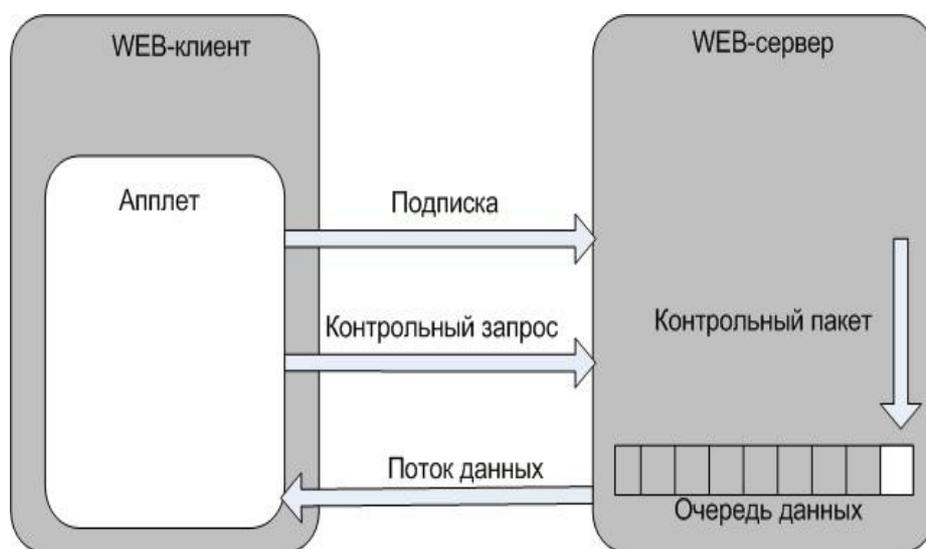


Рис. 2.

Время доставки данных T представляется:

$$T = T_1 + T_2 + T_3 + T_4 . \quad (1)$$

Здесь T_1 – время между событием и получением пакета данных сервером данных,

T_2 – время между получением данных сервером данных и приходом данных на WEB-сервер,

T_3 – время прохождения данных между WEB-сервером и клиентом,

T_4 – время обработки данных на WEB-сервере.

Промежуток времени T_3 возникает в результате использования WEB-технологий. Именно эта составляющая определяет пригодность WEB-технологий для доставки оперативных данных.

С частотой в контрольный промежуток времени T_c на WEB-сервер посылается команда, после получения которой WEB-сервер вставляет контрольный кадр в очередь событий с измененными данными. Апплет фиксирует время посылки команды и время получения ответного контрольного кадра. Разница во времени отправки команды T_s и времени получения контрольного кадра T_r апплетом дает оценку времени доставки данных T_3 .

$$T_3 = (T_r - T_s)/2 . \quad (2)$$

Время T_3 может отображаться в окне апплета. Если к моменту посылки новой контрольной команды не получен ответ на предыдущую команду ($2 \cdot T_3 > T_c$), апплет может выдать тревожное сообщение. Поскольку данные посылаются под видом единого WEB-ресурса, TCP/IP-протокол обеспечивает правильную нумерацию кадров, поэтому получение ответного контрольного кадра дает гарантию доставки всех предыдущих кадров с обновленными данными.

Длина очереди с данными на WEB-сервере равна

$$L = \sum_1^n P_i - \sum_1^k P_j , \quad (3)$$

где P_i – длины пакетов с данными, полученные WEB-сервером, а P_j – длины пакетов, отправленных подписчикам. Если количество полученных пакетов будет расти быстрее, чем количество отправленных пакетов (что теоретически возможно, так как WEB-сервер и сервер данных находятся в одной локальной сети), то WEB-сервер может израсходовать всю доступную память. Чтобы такого не произошло, в случае, если количество пакетов в очереди превышает некое,

наперед заданное число l , каждый новый входящий пакет с данными проверяется на соответствие пакетам, уже стоящим в очереди. Если в очереди существуют пакеты данных с такой же сигнатурой, то последний из этих пакетов заменяется на новый пакет, если пакетов с такой же сигнатурой нет, то новый пакет добавляется в конец очереди. Таким образом, длина очереди не может превышать предельное значение:

$$L \leq \sum_{i=1}^l P_i + \sum_{k=1}^M P_k, \quad (4)$$

где M – максимально возможное число различных сигналов, на которые может подписаться WEB-сервер.

Для поиска нужного пакета в очереди используются таблицы прямого доступа. Время нахождения пакета в очереди T_4 удовлетворяет выражению:

$$T_4 \leq \left(\sum_{i=1}^l P_i + \sum_{k=1}^M P_k \right) \Delta t, \quad (5)$$

где Δt – характерное время отправки одного байта. Другими словами, если длина очереди ограничена, то и время нахождения пакета в очереди тоже ограничено.

Проблема обработки данных может возникнуть и на стороне клиента, если апплет, отображающий данные, не будет успевать их отрисовывать. Для того чтобы этого не случилось, обработка входящих данных и их отрисовка реализована асинхронно в двух потоках.

Рассмотрим задачу прохождения оперативных данных через Проxy серверами и FireWall. Эти программы следят за тем, чтобы адресату передавались запросы-ответы только разрешенных типов. Как правило, разрешенной является информация, относящаяся к стандартным протоколам передачи данных и стандартным номерам портов адресатов. Эти программы могут создавать препятствие для прохождения оперативных данных. Чтобы решить эту проблему, могут потребоваться дополнительные настройки работы прокси-серверов как на стороне клиента, так и на стороне сервера.

За обслуживание и безопасность сети предприятия может отвечать отдельное подразделение предприятия, а в отдельных случаях это подразделение может принадлежать и совсем другому предприятию. Согласование вопросов, касающихся безопасности сети, может вызвать серьезные затруднения из-за межведомственных противоречий и распределения ответственности, что в свою очередь осложнит внедрение системы или сделает его вовсе невозможным. Поэтому передача данных без специальных настроек и разрешений администраторов сети является очень важным моментом. Проблемы могут возникнуть как с админи-

страторами на стороне клиентов, желающих получать данные, так и на стороне сервера, поставляющего данные.

Если возникает необходимость передавать информацию по нестандартному порту или протоколу, требуется специальное разрешение администратора сети. Такая необходимость возникает, если для передачи событийно-изменяемых данных использовать приложение отличное от WEB-сервера, или использовать механизм сервлетов.

К сожалению, старая проверенная технология CGI не может здесь спасти положение, так как она не может растягивать свой ответ и отсылать информацию порциями по мере изменения данных, а лишь может сформировать ответ на запрос и закончить свою работу отсылкой этого ответа. Другими словами, она работает по схеме: один запрос – один ответ. Для механизма событийно-изменяемых данных нам требуется схема: один запрос и много ответов. Для того чтобы механизм передачи данных мог работать на том же порту, что и WEB-сервер, он должен принадлежать процессу WEB-сервера. Поэтому для передачи оперативных данных необходимо встроить этот механизм в ядро стандартного WEB-сервера, либо реализовать собственный специализированный WEB-сервер.

Для того чтобы WEB-сервер мог передавать событийно-изменяемые оперативные данные клиентам, он должен уметь принимать эти данные от сервера данных.

Во второй главе рассмотрены средства графического отображения оперативных данных. Рассмотрен язык описания мнемосхем на базе XML, а также представлены различные способы отображения оперативных данных. Пример мнемосхемы показан на рис.3

Различные SCADA, разработанные крупными иностранными фирмами, такими как Intellution, WonderWare, FactoryLink, AdAstra, TraceMode и др., имеют как собственные средства создания мнемосхем, так и собственные средства их отображения. Поэтому фирмы-разработчики считают, что способы представления и хранения мнемосхем являются их внутренним делом. Форматы хранения мнемосхем, как правило, не описаны и являются закрытыми. Мнемосхема, созданная в одной среде разработки, не может быть использована в другой. Частично такая ситуация обусловлена жесткой конкуренцией на рынке SCADA, а частично – отсутствием соответствующих стандартов.

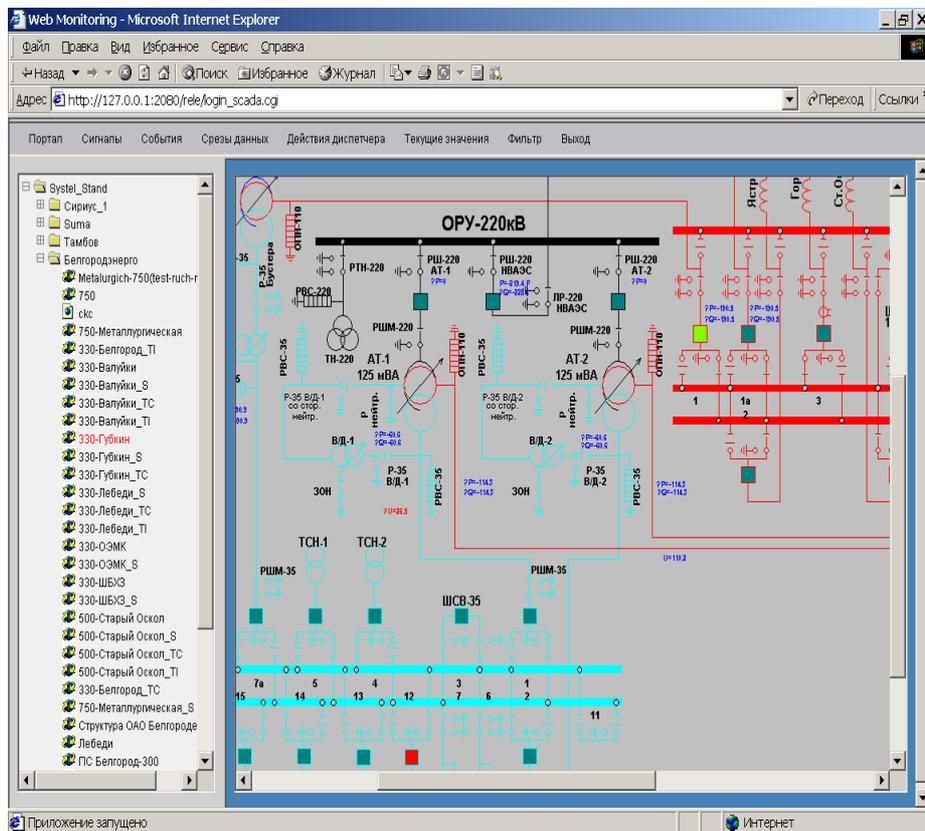


Рис. 3.

Задача четкой спецификации описания мнемосхем встала с расширением технологической базы, разрабатываемой SCADA “Систел” в область Интернет. Мнемосхемы, подготовленные в среде специализированного графического редактора, следовало отобразить с помощью новых средств. Для этого необходимо было разработать спецификацию представления мнемосхем, понятную различным приложениям, написанным на различных языках программирования и в различных операционных системах.

Описание мнемосхемы должно содержать графическое изображение и сценарий поведения мнемосхемы в случае тех или иных событий. События могут воз-

никать либо в результате интерактивных действий пользователя, либо в результате получения оперативных данных. Растровые изображения могут быть включены в мнемосхему как отдельные элементы, однако брать растровое изображение в качестве основного нежелательно, так как оно занимает большой объем памяти, а нам предстоит передавать описания мнемосхем по сети. Поэтому взята ориентация на векторную графику. Для описания изображений хотелось бы использовать распространенный стандартизированный формат.

Попытки использования стандартизированных средств для изображения мнемосхем предпринимаются на протяжении всей истории развития SCADA. Они обусловлены развитой средой разработки и отображения стандартных форматов, большим количеством различных профессионально исполненных графических редакторов и программ просмотра, которые освоены и привычны пользователям. Однако все такие предыдущие попытки постигла неудача. Причиной неудач является специфика протокола передачи оперативных данных, а также различия в конфигурационных базах данных в различных SCADA, что мешает делать привязку оперативных данных к изображению. Поэтому на базе XML был разработан новый язык описания мнемосхем.

Существует два принципиально разных подхода к разбору файлов XML, это DOM- и SAX-технологии. DOM – объектно-ориентированный подход, позволяющий работать с тэгами XML как с объектами. SAX – событийно-ориентированный подход, SAX-парсер вызывает определенные события при чтении тэгов XML. DOM – мощная и удобная техника работы, позволяющая модифицировать XML-файл, но требует большой объем памяти компьютера, так как все тэги и атрибуты тэгов реализовываются в виде объектов и хранятся в оперативной памяти. SAX требует минимальных ресурсов памяти, но при сложной организации XML-файла требует многократного прохода по нему для его разбора. Для разбора описаний мнемосхем было принято решение ориентироваться на SAX-технологии. Ориентация на SAX накладывает на язык описания мнемосхем требования по удобству и скорости разбора, так как некоторые мнемосхемы могут включать в себя сотни тысяч графических примитивов и их атрибутов.

Для обычных графических метафайлов проблем со структурой файла не возникает. Все графические элементы метафайла хранятся последовательно. Если метафайл создается на базе XML, то естественно графическому элементу метафайла поставить в соответствие XML-тэг, а атрибуты графических элементов записать в виде атрибутов соответствующего XML-тэга. Однако описание мнемосхемы отличается от обычного метафайла тем, что кроме графического изображения в нем присутствуют сценарии поведения схемы в случае тех или иных событий, привязка состояния оперативных данных к графическим атрибутам либо координатам графических элементов. Для эффективной работы с описаниями

мнемосхем необходимо придумать подходящую систему базовых понятий языка описания.

Отправной точкой является метафайл, реализованный на основе XML. Первый шаг по оживлению метафайла – это добавление к графическим элементам элемента Action. Элементу Action будет соответствовать XML-тэг, описывающий действие сценария мнемосхемы в результате возникновения какого-либо события. Действием может быть изменение видимости элемента, цвета, координат, начало или прекращение мерцания одного из графических элементов, переход на другую мнемосхему, масштабирование и т.п. Элемент Action избавит нас от необходимости использовать JavaScript для обработки событий. Ключевыми понятиями становятся: ACTION – действие, PRODUCER – элемент, который инициирует действие, TARGET – элемент, на который действие направлено. Событие EVENT, в результате которого происходит действие, будет являться атрибутом объекта PRODUCER. В результате имеем связку PRODUCER-ACTION-TARGET.

```
<Button position = "-261;228;-446;269" color = "00ff00" Title = "TEST">
  <Producer event = "MouseUp">
    <Action type = "change_color" value = "ff0000" >
      <Target Object = "Rect" ID = "Rect_ID_1" />
    </Action>
  </Producer>
</Button>
```

Так приведенный фрагмент кода описывает кнопку с названием “TEST”, которая перекрашивает прямоугольник с идентификатором "Rect_ID_1" в красный цвет. При линейном просмотре файла, для того чтобы установить связь между кнопкой, производящей действие, и прямоугольником, на который это действие направлено, необходимо, чтобы прямоугольник был уже определен выше по файлу. Порядок графических элементов определяет порядок их рисования, который влияет на изображение мнемосхемы. Поэтому может оказаться, что кнопка расположена выше, чем прямоугольник, и в момент описания кнопки не будет найден идентификатор прямоугольника. В этом случае можно изменить связку PRODUCER-ACTION-TARGET на TARGET-ACTION-PRODUCER

```
<Rectangle position = "-60.0;-50.0;40.0;50.0" color = "c0c0c0">
  <Target>
    <Action nick = "change_color" value = "0000ff" >
      <Attach producer = "Button" id="Button_ID_1" event =
"MouseUp" />
```

```

</Action>
</Target>
</Rectangle>

```

Таким образом, можно остаться в рамках структуры XML при описании событий и устранить необходимость жесткой привязки идентификаторов к графическим элементам. В нашем случае идентификатором может являться обычный порядковый номер элемента, что в свою очередь снимает необходимость отслеживания уникальности идентификаторов и ускоряет разбор и создание описаний мнемосхем. Для каждого действия должен существовать элемент PRODUCER и элемент TARGET. Такие действия как приход оперативных данных, открытие нового окна или меню, могут не иметь элемента PRODUCER или элемента TARGET, в таком случае будет использоваться в качестве недостающего элемента сама схема, или объект главного окна.

Часто возникает потребность производить действия не над одним, а над целым классом графических элементов. Например, надо показать на схеме только объекты определенного назначения, все остальные объекты должны быть скрыты, или необходимо выделить объекты определенного типа новым цветом. Для этого все графические элементы на схеме должны быть сгруппированы соответствующим образом.

```

<Group name="Layer1" Range = "-67;-360;67;360" Transform ="1;0;0;0;-1;0" Port
="-670;-360;670;360" >
....
</Group>

```

Над координатами группы элементов можно выполнять аффинные преобразования. Параметры матрицы аффинного преобразования для группы задаются с помощью атрибута тэга группы Transform= "a₁,a₂,a₃,a₄,a₅,a₆"

$$\begin{pmatrix} x \\ y \\ 1 \end{pmatrix} = \begin{pmatrix} a_1 & a_2 & a_3 \\ a_4 & a_5 & a_6 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x \\ y \\ 1 \end{pmatrix}. \quad (6)$$

Структура XML, основанная на вложенности тэгов, не может обеспечить группировку элементов по нескольким признакам, поэтому для группировки по признакам необходимо добавлять признак внутрь тэгов элементов.

```
<Rectangle position = "-60.0;-50.0;40.0;50.0" color = "c0c0c0">  
  <Feature name="technical">  
</Rectangle>
```

Описание мнемосхемы на уровне графических элементов является низкоуровневым. Если на мнемосхеме нарисован трансформатор, то ничего нельзя сказать о свойствах этого трансформатора, нельзя выполнять действия с ним как с единым целым, нельзя тиражировать его на мнемосхеме. Для этого необходимо ввести объекты более высокого уровня, которые должны иметь свое типовое графическое представление и наборы свойств. Приведем пример такого объекта:

```
<Object Name = "Lamp" Range="-100;-100,100,100">  
  <Property name = "Voltage" type="integer" value="150">  
  <Property name = "Type" type="string" value="daylight">  
  <Line position="-80;-80;80;80" color="000000" />  
  <Line position="-80;80;80;-80" color="000000" />  
  <Ellipse position="-80;-80;80;80" color="000000" fill="no" />  
</Object>
```

Описание объекта задается в своих собственных координатах и определяет список свойств объекта. После определения объекта он может использоваться в файле описания мнемосхемы как графический элемент. При рисовании на мнемосхеме объект масштабируется в соответствии с заданными координатами.

```
< Lamp position = "-60.0;-50.0;40.0;50.0" />
```

К свойствам объекта могут быть привязаны графические действия, таким образом, меняя свойства объекта, можно изменять его изображение на мнемосхеме. На базе объектов более высокого уровня могут быть созданы библиотеки стандартных элементов.

В третьей главе рассмотрена организация приложения для мониторинга оперативных данных, показано место системы WEB-мониторинга на энергопредприятии. Здесь также обсуждаются технические аспекты реализации авторизации и персонального доступа пользователей к информационным ресурсам системы WEB-мониторинга. Обсуждается вопрос автоматического сохранения пользовательской информации, заполненной им в виде HTML-форм.

Широко распространенным решением для хранения данных между рабочими сессиями клиента стали базы данных. Без баз данных сегодня не обходится почти ни один WEB-проект. Интернет-магазины, различные форумы, электронные библиотеки, почтовые ящики – все они используют базы данных для хране-

ния своей информации. Базы данных удобны для хранения информации, предоставляемой пользователю, а для хранения информации пользователей и о пользователях они удобны только тогда, когда эта информация строго определена. Действительно, для каждого элемента пользовательской информации должен быть заведен соответствующий столбец в какой-либо таблице базы данных и приведен в соответствие механизм записи и чтения этих данных. В случаях, когда отдельные элементы пользовательской информации имеют значение лишь для одного конкретного пользователя или когда информация разнообразна и часто меняет свое структурное содержание, хранение такой информации в базе данных становится трудоемкой или просто невозможной задачей.

Задачу хранения такой разнородной информации можно легко решить, если соединить вместе технологию работы с конфигурационными файлами и подстановку пользовательских переменных в HTML-документ.

Например, создается некая HTML-форма с именем MySettings.html, в которой расположены различные элементы ввода, которые пользователь может заполнить по своему усмотрению. Допустим, внутри этой формы расположен текстовый элемент ввода:

```
<input type = "text" name = "MyColor" value = "<%= MyColor%>">
```

Расширение файла MySettings.html меняется с html на frm. Это новое расширение будет сообщать WEB-серверу, что с данной формой (перед ее отправкой пользователю) следует проделать предварительную обработку, а именно, необходимо подставить переменные из конфигурационного файла в соответствующие поля формы. Имя этого файла состоит из названия формы и имени пользователя, если имя пользователя Guest, то имя конфигурационного файла будет MySettings_Guest.cfg. Внутри этого файла должна найтись строка, которая соответствует переменной подстановки <%= MyColor%>:

```
<MyColor value = "#ffffff">
```

Соответствующее значение MyColor будет подставлено в файл MySettings.frm перед его отправкой пользователю. То же самое происходит при подтверждении формы (submit), только в обратном порядке. При подтверждении формы, согласно протоколу http, на сервер вместе с запросом присылаются в качестве параметров все поля формы. Перебираем в цикле все присланные поля и создаем новый конфигурационный файл MySettings_Guest.cfg с переменными одноименными полям формы.

Используя описанный механизм сохранения форм, можно не заботиться о создании полей в базах данных для хранения пользовательской информации и

о функциях, производящих запись и чтение этой информации. Данный механизм достаточно реализовать в WEB-сервере один раз, после чего он сможет работать с произвольными формами, имеющими различные наборы элементов данных.

Механизм сохранения форм оказался очень удобным, он используется также для генерации диспетчерских отчетов.

В заключении сформулированы основные результаты работы:

1. Разработано программное обеспечение для отображения в Интернет оперативной диспетчерской информации, а также архивных данных.
2. На базе XML разработан специализированный язык описания мнемосхем, отображающих оперативные данные.
3. Предложен механизм контроля качества доставки оперативных данных через Интернет.
4. Предложен ряд интересных с точки зрения философии WEB-технологий решений, например сохранения клиентской информации на стороне сервера в виде конфигурационных XML-файлов.
5. Решена проблема прохождения оперативных данных через Proxy-сервера и Firewall.
6. На основе эксплуатации системы на важных энергопредприятиях России показана пригодность WEB-технологий для мониторинга оперативных данных.

Список литературы

1. Публикации в изданиях, рекомендованных ВАК РФ:

- [1] С.А. Сухов, В.И. Ухов. *Подходы к созданию АСУТП на основе WEB-технологий.* // Автоматизация и Современные Технологии. N10, 2006.
- [2] С.А. Сухов. *Мониторинг оперативных диспетчерских данных средствами Интернет.* // Автоматизация и Современные Технологии. N3, 2007.

2. Другие публикации:

- [3] С.А. Сухов. *Технологические аспекты передачи оперативных данных в среде Интернет.* // Вестник университета Дубна. N2, 2005. с. 12-17.
- [4] С.А. Сухов, Н.Ю. Кульман. *Создание описания мнемосхем на основе XML.* // Вестник университета Дубна. N1, 2006, с. 52-58.
- [5] С.И. Купцов, В.Н. Пелешко, С.А. Сухов, В.И. Ухов. *Развитие автоматизированной системы радиационного контроля ускорителя ИФВЭ в рамках современных информационных технологий.* // XVIII Конференция по ускорителям заряженных частиц. RUPAC-2002. Сборник докладов том 2 с. 636-641.
- [6] С.Н. Рыкованов, В.И. Виноградов, В.И. Ухов, С.А. Сухов, Д.А. Хабибуллин. *Терминальная система сбора и отображения данных в Интернет.* Международный симпозиум по проблемам модульных систем и сетей ICSNET' 2001.

Рукопись поступила 24 сентября 2007 г.

С.А. Сухов
Разработка системы мониторинга оперативных данных на основе
WEB-технологий.

Редактор Л.Ф. Васильева.

Подписано к печати 25.09.2007. Формат 60 × 84/16. Офсетная печать.
Печ.л. 1,16. Уч.–изд.л. 1,3. Тираж 100. Заказ 78. Индекс 3649.

ГНЦ РФ Институт физики высоких энергий,
142281, Протвино Московской обл.

Индекс 3649

А В Т О Р Е Ф Е Р А Т 2007-15, ИФВЭ, 2007
