УДК 621.311.153.001.24

INTERNET-ТЕХНОЛОГИИ В ЭНЕРГЕТИКЕ

Апухтин А.С., Джура С.Г.
Донецкий государственный технический университет
djura@pandora.kita.dgtu.donetsk.ua

An overview of the Internet-technologies is presented. A wide range of Internet-methods have been reviewed and summarired in this article. In clause the classification of existing Internet-technologies is given in power. The analysis of existing problems concerning distributed is given calculations, expert systems, systems of support of acceptance of the decisions, intellectual databases, PUSH-technologies, relational searches.

Среди перспективных направлений развития Internet можно выделить несколько технологических концепций: одни из них уже сейчас начинают широко использоваться в энергетических приложениях, другие еще не вышли из стадии разработки. Но всех их объединяет одно: довольно скоро они сыграют значительную роль в дальнейших трансформациях Internet, существенно расширяя возможности ее пользователей. С начала 90-х годов ничто так не привлекает внимания в мире компьютерных технологий, как Internet. Возникшая из студенческой дипломной работы и ставшая несомненным "хитом" уходящего века, она далеко не исчерпала возможности своего развития. Грандиозность Internet, ее необычайно быстрый рост по сравнению с другими технологиями внушают уважение и гордость за компьютерную отрасль, науку и за человечество вообще. Но еще больше поражают перспективы развития Сети, обещающие уже в ближайшем будущем во многом изменить нашу жизнь. В связи с вышесказанным возникают новые возможности для решения старых проблем. Одними из таких проблем для энергетики являются:

1. Проблема распределенных вычислений.
2. Проблема создания и пополнения распределенных баз данных (РБД).
3. Проблема реляционных средств поиска в РБД.
4. Проблема защиты информации.

Данная статья направлена на поиск новых путей решения первых трех указанных выше проблем (поиски решения четвертой - тема отдельной статьи, хотя она несомненно связана с остальными). Что же понимается под распределенными базами данных? Это единая база данных, но расположенная на разных машинах одновременно, а значит, это база с сетевыми возможностями. Сетевые возможности позволяют практически неограниченно повышать вычислительную мощь пользователя за счет вновь присоединяемых машин. Зачем нужны такие базы? Для того, чтобы несколько пользователей могли одновременно ею пользоваться и оперативно ее пополнять. Последний момент очень важен. Дело в том, что конструкторские базы данных выступают как средство аккумуляции опыта целых поколений проектировщиков и при оперативном пополнении и использовании такой базы могут пользоваться все большее количество пользователей и их опыт (а стало быть и качество конечного продукта) за счет такого инструмента растет в геометрической пропорции. Только настоящий прогресс в вычислительной технике сделал возможным реально сегодня начать воплощать указанные проекты в жизнь. При этом надо оговориться, что сетевые продукты сейчас весьма дороги, но при правильной постановке вопроса это решаемая задача [1]. Рассмотрим состояние Internet - технологий в энергетике. Последние можно классифицировать следующим образом:

1. WWW-технология (World Wide Web). Под ними будем понимать расположение на WEB-серверах различной информации (реклама, проекты, инвестиционный предложения и т.д.) в активном режиме (on-line).
2. FTP-технология (File Transport Protokol). Под этим будем понимать расположение информации в пассивном режиме на FTP-серверах.
3. ICQ-технология. Фактически это технологии быстрого обмена данными не только двух одновременно, а и ряда пользователей в режиме реального времени (фактически Internet-пейджер).
4. EMAIL-технология. Это технология обмена электронными письмами очень актуальная для удаленных пользователей.
5. NET-технологии. Под этим будем понимать автоматизированные рабочие места конструктора-разработчика, распределенные системы управления базами данных (СУБД) и реляционные средства поиска. Реляционные средства означает средства дословно "отношений", то есть сложных запросов, разработанных для языков искусственного интеллекта. В конечном итоге это взаимодействие сети сетей Internet (как внешней объединяющей сети) и конечного ряда Intranet (как внутренних корпоративных сетей).
6. Комплексные и интегрированные сетевые технологии.
   - 6.1. Экспертные системы.
   - 6.2. Системы поддержки принятия решений.
   - 6.3. Интеллектуальные базы данных.
7. Internet-телефония и прием-пересылка факсов.

Реально все перечисленные технологии должны работать в параллельном режиме на целом ряде машин одновременно [2]. Целый ряд вопросов уже решен, однако остановимся на том, что еще ждет своего решения, а именно целый ряд описанных выше проблем. Какими видятся их решения для электроэнергетики?

Рассмотрим первую из указанных проблем: (распределенные вычисления в энергетике). Распределенные вычисления в энергетике можно использовать в следующих вариантах:

1. Общие конструкторские базы данных для проектирования как элементов так и комплексных проектов в электроэнергетике, которые интегрированы с автоматизированными рабочими местами (АРМ) конструкторов;

2. Экспертные системы для анализа электроэнергетических проектов;

3. Базы знаний по электроэнергетическим направлениям;

4. Системы поддержки принятия решений как для оперативного персонала (диспетчеризация и т.д.) так и для управляющего звена (министерства, ведомства).

Каким видятся схемы таких вычислений? Одну из них приведем ниже (см. рис.1).

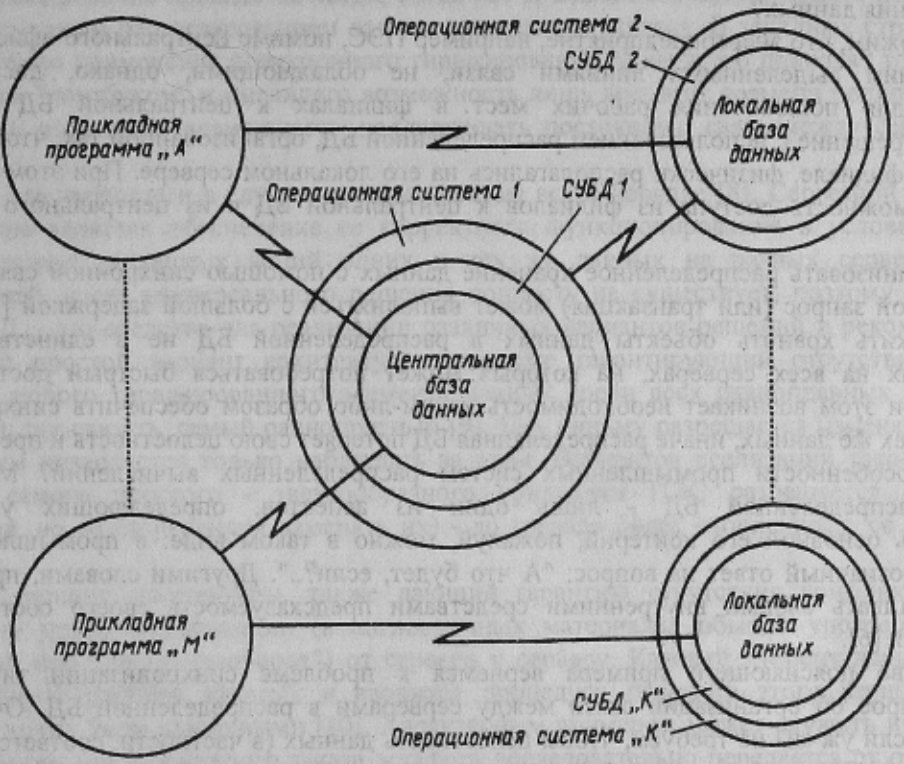

Рисунок 1 – взаимодействие центральной и локальной баз данных

Это общая схема. Рассмотрим более подробно локальную базу данных (назовем ее системой управления микробазой данных).

Использование СУБД не ограничивается центральной базой данных системы. Ее средства должны быть доступны пользователям, работающим на всех АРМ [3]. В связи с этим приходится рассматривать концепцию системы управления микробазой данных (СУМБД).

При возникновении аварийной ситуации, требующей повторения доступа к данным, система должна иметь возможность возобновить обработку транзакции с некоторого шага, предшествовавшего сбою, с восстановлением соответствующего состояния базы данных, что весьма актуально для сетей стран СНГ, где качественная связь только вводится.

В процессе функционирования системы должны быть решены задачи, которые можно классифицировать следующим образом:

1. Стандартизация (здесь сталкиваются корпоративные интересы и интересы различных групп и даже фирм разработчиков);

2. Пополнение распределенных баз данных и знаний (нежелание делиться своими наработками);

3. Администрирование разработанных продуктов (поддержание в работоспособном состоянии, оперативное устранение неполадок, возникающих при эксплуатации);

4. Интеграция старых АРМ (написанных на старом программном обеспечении) в новую систему.

5. Согласование работы единой системы распределенной архитектуры.

Далее рассмотрим процесс функционирования самой распределенной базы данных, а на ее примере будут видны основные направления построения других указанных выше типов программного обеспечения.

Рассмотрим распределенные вычисления на лучшей (по многим оценкам) из распределенных баз данных (БД) Oracle [4]. Напомним, что самый простой вариант распределения данных - тот, при котором любой объект распределенной БД хранится только в одном экземпляре. Это означает, что все серверы, затронутые распределенной (одновременно изменяющей данные, хранящиеся на нескольких серверах) транзакцией, должны взаимодействовать в синхронном режиме. Для этого нужны надежные, постоянно доступные и достаточно быстрые линии связи между серверами.

Если эти условия не соблюдены, то можно растиражировать объекты данных на всех серверах, где к ним может потребоваться быстрый доступ. Самым простым вариантом тиражирования в Oracle являются так называемые неизменяемые снимки (read-only snapshots). Они определяются так же, как представления - view, т.е. могут быть основаны и на нескольких таблицах, доступны только для чтения и обновляются по заданному сценарию и расписанию. Более сложными являются изменяемые снимки, предоставляющие возможность модификации удаленных копий. При этом один из серверов является владельцем "оригинала" данных. Последний атомарный вариант - "тиражирование с множественными хозяевами" (multi-master site replication), когда полностью тиражируются целые наборы объектов БД. В них помимо таблиц могут входить индексы, представления, триггеры, хранимые процедуры, синонимы, генераторы последовательностей. Любые изменения тиражированных данных непосредственно распространяются среди всех хозяев. Как это применимо для тиражирования данных?

Предположим, что энергопредприятие, например ПЭС, помимо центрального офиса имеет ряд филиалов, связанных с ним выделенными линиями связи, не обладающими, однако, достаточной пропускной способностью для подключения рабочих мест в филиалах к центральной БД в режиме клиентов. Напрашивается решение с использованием распределенной БД, организованной так, чтобы данные, чаще всего требующиеся в филиале, физически располагались на его локальном сервере. При этом возникает вопрос: как обеспечить возможность доступа из филиалов к центральной БД и из центрального офиса - к данным в филиалах?

Если организовать распределенное хранение данных с помощью синхронной связи без тиражирования данных, то любой запрос (или транзакция) может выполняться с большой задержкой [5]. В качестве выхода можно предложить хранить объекты данных в распределенной БД не в единственном экземпляре, а тиражировать их на всех серверах, на которых может потребоваться быстрый доступ к этим объектам. Естественно, при этом возникает необходимость каким-либо образом обеспечить синхронизацию различных копий одних и тех же данных, иначе распределенная БД потеряет свою целостность и превратится в хаос.

Каковы особенности промышленных систем распределенных вычислений? Методика обеспечения целостности распределенной БД - лишь один из аспектов, определяющих упомянутое различие. Сформулировать основной его критерий, пожалуй, можно в таком виде: в промышленной системе всегда можно дать однозначный ответ на вопрос: "А что будет, если?..". Другими словами, промышленная система должна обеспечивать своими внутренними средствами предсказуемость своего состояния независимо от внешних условий [6].

В качестве поясняющего примера вернемся к проблеме синхронизации тиражируемых данных. Рассмотрим вопрос об организации связи между серверами в распределенной БД. Очень соблазнительной является идея: если уж мы не требуем, чтобы целостность данных (в частности, соответствие тиражированных копий друг другу) обеспечивалась в любой момент времени, нельзя ли организовать указанную связь на основе электронной почты?

На первый взгляд, ничего опасного в этом нет, но вспомним, что электронная почта не гарантирует, что "письма" будут получены адресатом в той же последовательности, в которой они были посланы, или что все они вообще будут получены. Даже если с помощью специального контроля на приеме последствия данной неоднозначности сводятся к минимуму, все равно невозможно предсказать момент времени, когда информация об изменении одной из копий объекта дойдет до других его копий, и в каком состоянии к этому моменту будет находиться изменившаяся копия.

Означает ли это, что электронной почтой в распределенной БД пользоваться вообще нельзя? И да, и нет. Все зависит от требований к системе. Если "свежесть" тиражированных данных требуется в пределах суток, а последствия непредсказуемости системы и потери ее целостности не слишком разрушительны, то почему бы и нет? Но если все-таки требуется действительно промышленная система, то необходимо очень тщательно оценить все возможные последствия применения выбранного механизма взаимодействия и ввести в использование того же тиражирования такие ограничения, которые обеспечили бы поддержку требуемого уровня целостности системы.

Таким образом, при проектировании системы с тиражируемыми данными необходимо как минимум задать себе следующие вопросы:

- Допустимо ли временное рассогласование тиражированных данных?
- Если да, то в каких временных пределах должна осуществляться процедура их согласования и каким образом она должна инициироваться?
- Не могут ли привести сбои каких-либо элементов системы к безвозвратной потере целостности распределенной БД?
- Какова должна быть дисциплина работы с тиражированными данными, чтобы исключить возможность конфликтов между модификацией различных копий одних и тех же данных, или - если такие конфликты допускать - какова должна быть процедура их разрешения?

В принципе возможны два режима тиражирования [7]: синхронный, когда все изменения данных распространяются немедленно, и асинхронный, когда по отношению к этим изменениям применяется алгоритм "запомнить и передать" (store and forward), а момент передачи выбирается по заданному правилу (или явно инициируется, например, когда восстанавливается связь).

Синхронный вариант оправдан, по-видимому, для расчетов за энергоносители или услуги электроэнергетических предприятий. Иначе, в случае асинхронного тиражирования, появляется опасность, что нечистый на руку клиент энергопредприятия может, к примеру, несколько раз закрыть свой счет, быстро перемещаясь из филиала в филиал. Впрочем, как показано ниже, данная проблема может быть разрешена и по-другому.

По сравнению с синхронной связью без тиражирования, преимущества асинхронной связи состоят в том, что, во-первых, запросы выполняются на локальном сервере и не требуют передачи данных между серверами, во-вторых, транзакции, хотя и требуют распространения изменений, все же выполняются для клиентов быстрее, поскольку это распространение происходит асинхронно по отношению к транзакциям и клиент не ждет окончания обмена данными между серверами.

Асинхронное тиражирование применимо тогда, когда нет возможности поддерживать постоянную связь между серверами, а временным рассогласованием данных можно поступиться. В западных странах в качестве примера, иллюстрирующего применение асинхронного тиражирования, чаще всего приводят торгового агента, разъезжающего со своим "ноутбуком" и имеющего возможность лишь время от времени подключаться к сети. В наших условиях, увы, отсутствие возможности поддерживать постоянную связь между серверами весьма типично.

Впрочем, задача эта непроста и в случае, когда со связью все в порядке [8]. Серьезной проблемой при проектировании системы является обеспечение ее корректного функционирования в условиях возможных конфликтов по обновлению различных копий одних и тех же данных на разных серверах. Здесь мы сталкиваемся с ситуацией, когда универсального решения попросту не существует, поэтому все, что может предоставить вам СУБД, - это средства для реализации различных вариантов решений и рекомендации по их использованию. Самый простой вариант архитектуры, заведомо гарантирующий отсутствие конфликтов, предполагает, что для любого тиражированного элемента данных среди всех равноправных хранителей его копий выбирается один, так сказать, самый равноправный [9]. Ему одному разрешается изменять этот элемент данных, всем остальным позволяется только наблюдать за этим. Вариантов реализации такой схемы может быть множество: от самого простого - звездообразного (филиалам ПЭС разрешается видеть данные центрального отделения, но не разрешается изменять их) - до гораздо более изощренных, со сложной сетью неизменяемых снимков.

Более сложный вариант архитектуры, также дающий гарантию отсутствия конфликтов, разрешает динамическую передачу права модификации (в англоязычных материалах обычно употребляется термин, буквально переводимый как "документооборот") от сервера к серверу. Каждый элемент данных снабжается специальным атрибутом "разрешена запись", и вводится процедура передачи этого атрибута. Варианты реализации опять-таки могут быть различными [10]. Характерным примером может служить информационная система энергопредприятия, где для каждого заказа эстафета последовательно передается от отдела продаж к управлению ресурсами и далее к отделу доставки. Аналогичная схема в принципе может решить проблему "многократного закрытия счета" в банке энергопредприятия, упомянутую выше.

Любая другая организация архитектуры тиражирования допускает возникновение конфликтов, следовательно, не остается ничего другого, как пытаться их разрешать. От СУБД здесь в первую очередь требуется обеспечить автоматическое детектирование конфликтов и возможность извещения о них (излишне, видимо, упоминать, что Oracle делает это), ибо момент возникновения конфликта зависит от механизма распространения изменений [11]. Но просто извещать о проблеме и ждать ее "ручного" разрешения, наверное, не лучший выход. По возможности, нужно стараться разрешать конфликты автоматически.

Oracle предлагает с этой целью набор процедур разрешения конфликтов, которые можно ассоциировать с группами столбцов таблицы. Руководствоваться при этом приходится обыкновенным здравым смыслом: скажем, если речь идет об описательных данных клиента, в качестве разрешающей функции разумно выбрать последнюю по времени модификацию, изменения последнего показания счетчика и имеет смысл просуммировать и т.д.

Помимо внушительного набора стандартных функций разрешения конфликтов [12] можно использовать и свои собственные. Однако не все так просто. Функции разрешения конфликтов должны обеспечить конвергенцию данных, то есть их непременную установку в одно и то же значение. Полная же конвергенция не всегда возможна, особенно в случае, когда в конфликте участвует более двух серверов [13]. Если используется нестандартная функция, для определения ее свойств может потребоваться весьма тонкий анализ, а для стандартных он уже проведен. Как бы то ни было, в системе необходимо предусматривать либо выбор только тех функций, которые обеспечивают полную конвергенцию данных при принятой дисциплине доступа к ним, либо использовать извещение администратора, когда функция не способна справиться с ситуацией самостоятельно.

Поддержка резервной копии базы данных Oracle предлагает еще один механизм, напоминающий тиражирование. Это предназначенная для повышения устойчивости системы к сбоям поддержка резервной копии БД (standby database). Смешивать данный механизм с тиражированием, пожалуй, не стоит, так как резервная БД недоступна для пользователей одновременно с основной. Зато отсутствует дополнительная нагрузка на ядро основного сервера, связанная с указанным выше процессом. Собственно, резервная БД как раз

и функционирует в режиме перманентного восстановления, используя журнальные файлы основной БД, переданные тем или иным способом на резервный сервер [14].

Очень важной является возможностях использования Oracle в гетерогенных распределенных БД. Выбор варианта решения задачи во многом зависит от составляющих гетерогенную систему СУБД.

Если это реляционные СУБД (MS SQL Server, Informix, Sybase, DB2, CA Ingres), можно использовать так называемые "прозрачные" шлюзы для объединения их с Oracle. Для пользователя такого шлюза полностью имитируется функциональная среда сервера Oracle при доступе к данным, хранящимся в "чужих" СУБД.

Для реализации шлюза фактически используется промежуточный сервер Oracle, за счет которого и достигается эффект "ораклизации" данных. Чаще всего он функционирует на том же компьютере, что и "чужой" сервер. Например, если пользователь вызывает хранимую процедуру на PL/SQL, то она фактически выполняется сервером-шлюзом (СУБД других производителей с PL/SQL не работают), а "чужому" серверу передаются только SQL-предложения, содержащиеся или сформированные в процедуре.

Сложнее обстоит дело, если необходимо получить доступ к данным, хранящимся в нереляционных СУБД (ADABAS, VSAM и пр.) [15]. В таком случае, как правило, невозможно формально однозначно отобразить эти данные в реляционные структуры Oracle, поэтому подход прозрачных шлюзов не применим. Тем не менее Oracle может предложить решение для таких ситуаций в виде процедурных шлюзов. В них вместо стандартного SQL для взаимодействия с "чужими" данными предоставляется библиотека процедур, с помощью которых разработчик может реализовать необходимое отображение данных. Каковы же перспективы развития подобных вычислений?

Среди множества перспективных технологий, расширяющих возможности Internet, на сегодняшний день выделяется несколько связанных технологических концепций. Это интеллектуальные агенты - технология, направленная на расширение возможностей интеллектуального поиска информации в Internet; push-технология, упрощающая доставку больших объемов мультимедийных данных, а также свежих новостей на пользовательские компьютеры. К этому списку можно добавить также RPSS-системы моделирования ролевых игр, обеспечивающие качественно новый способ общения через Internet.

Что касается доставки информации в реальном времени, то Push-технология уже сегодня представляет собой хорошо отработанную, интенсивно расширяющуюся область Internet-приложений [16]. С ее помощью пользователи могут не только направлять запросы на сервер в режиме Online, но также регулярно получать информацию по заранее размещенным запросам. В этом случае между сервером и клиентом устанавливается так называемый Internet-"канал" (channel), причем инициатором передачи данных может быть как клиент, так и сервер. При любых изменениях информации на сервере, в зависимости от способа "подписки" на данный канал, он либо оповещает программу-"приемник" о появлении новых данных, либо сразу пересылает изменения и дополнения по установленному набору подписных рубрик на компьютер клиента [17]. При этом не обязательно, чтобы на пользовательском компьютере был загружен браузер, - все необходимые действия выполняет небольшая клиентская программа - Tuner.

С точки зрения пользователя, механизмы Push-технологии просты и эффективны. Необходимо выбрать нужный Internet-канал, установить желаемый период и время суток для обновления информации, и готово! По мере дополнения свежие данные будут автомататически доставляться на ваш компьютер. Поскольку доставка информации производится в режиме download, то есть сетевой "подкачки", низкая производительность линии не является значительным препятствием даже при доставке мультимедийных данных большого объема.

Таким образом, пользователь получает прекрасную возможность не только ознакомится со свежей актуальной информацией по энергетике почти сразу после его публикации, но также постоянно быть в курсе текущих событий, получая сводки новостей от разных информационных источников непосредственно на свой ПК. Более того, используя ряд несложных средств, сегодня практически любая компания может организовать собственный канал, предназначенный для открытого или внутреннего "вещания".

На сегодняшний день существует три наиболее популярных средства, поддерживающих Push-технологи: Castanet Tuner фирмы Marirni Microsoft, Internet Explorer и NetCaster, входящий в состав Netscape Communicator 4.7 [18]. Фирма Marimba с Castanet Tuner стала пионером и автором основных стандартов Push-технологии. К сожалению, например, последняя версия 2.1 не работает на Proxy (такое случается иногда с новыми программами). Однако, следует отметить, что среди имеющихся производителей только Marimba предоставляет полный набор программного обеспечения, включая Push-сервер, позволяющий организовать собственный Internet- или Intranet-канал, использующий Push-технологию. Более того, решение Marimba позволяет заметно разгрузит информационный трафик корпоративной сети за счет кэширования Internet каналов на локальном сервере организации. Это дает возможность обеспечить коллективный доступ к часто используемым каналам, таким как PointCast или Science Daily, так что многомегабайтные объемы обновляемой информации размещаются на сервере организации, не засоряя локальные диски пользовательских компьютеров.

После выхода Internet Explorer 5.0 Microsoft, похоже, становится несомненным лидером в области Push-технологии. Концепция Active Desktop, определяющая "лицо" Windows 98, позволяет доставлять свежую информацию по подписке буквально на "рабочий стол" Windows. Установка системы, подписка на каналы (их предлагается свыше четырехсот!), настройка тюнера, - весь необходимый набор функций работы с каналами реализован просто и удобно.

Netcaster, несмотря на традиционно высокий уровень разработки, представляет по мнению многих обозревателей еще один пример утерянной инициативы Netscape в области Internet-браузеров. Удобное, продуманное средство со множеством возможностей, - но далеко отстает от MSIE. Несомненно, Netscape продолжит борьбу, но после выхода Windows 98 скандально известный Active Desktop, интегрирующий MSIE 4.0 в операционную систему, дает Netscape слишком мало шансов.

В последнее время многие компьютерные издания начинают говорить о "второй волне" push-приложений. Среди них можно выделить Headliner от BackWeb, TORSO от Wayfarer и целый ряд новых продуктов от малоизвестных пока производителей. Обозреватели считают, что в ближайшем будущем из "второй волны" могут вырасти новые гиганты, подобно тому, как это уже случилось с Microsoft.

В основе Push-технологии лежит борьба с информационной перегрузкой, ставшей в определенном смысле "болезнью" уходящего века. Этой болезни подвержены большинство магистралей, несмотря на постоянный рост их пропускной способности. Возникает знакомое любому сетевому администратору подозрение, что информация способна браться ниоткуда, заполняя любые каналы и носители. В результате развитие Push-технологии можно рассматривать только лишь как проблему "интеллигентного" обращения информации по сетям глобального и локального масштабов. И долгожданная компьютерная революция, связанная с построением действительно интеллектуальных систем, должна решить эту проблему.

Знаменитый лозунг, родившийся где-то внутри Sun Microsystems, гласит: "Сеть - это компьютер". Согласно другому высказыванию, принадлежащему известному математику и психологу Дэну Герцелю, "сеть - это интеллект". Действительно, сегодня Internet представляет собой большую и сложную систему, объединяющую миллионы узлов и связей, по которым транспортируется информация. Система эта сложна настолько, что, с точки зрения синергетики, способна, более того, призвана стать самостоятельным носителем интеллекта. Учитывая огромные информационные объемы и значительные вычислительные ресурсы в узлах сети, интеллектуальные способности этого "монстра" являются непрогнозируемыми.

Конечно, речь не идет о неких апокалиптических предостережениях. При отсутствии организующей идеи Internet может навсегда остаться хаотическим сборищем информационных потоков, интеллектуальным не в большей степени, чем телевизионная сеть. Вполне возможно, что, по крайней мере в течение нескольких ближайших десятилетий, Всемирная Сеть будет развиваться по интенсивному пути, характерному для компьютерной отрасли, уже не первое десятилетие бодро шагающей по проторенной фон-Неймановской дороге, и можно смело утверждать, что в 2001 ситуация не изменится (компьютер "образца 1998 года" построен по той же фон Неймановской схеме "потока команд", что и его предшественники начиная с 1946 года). А архитектуры пятого поколения, представленные машинами потока данных, программируемыми логическими матрицами, ассоциативной памятью и другими экзотическими технологиями, оказались просто нерентабельными (с коммерческой точки зрения).

Однако в сегодняшней Internet на фоне буйно расцветающей поросли рекламы и коммерции уже заметны ростки технологии, которая способна в будущем стать той самой организационной идеей, превращающей Сеть в Интеллект. Речь идет о технологии Software Agents, точнее, о ее более узкой области - Intelligent Agents (IA - интеллектуальные агенты). Вообще понятие IA становится все более популярным и вместе с тем более размытым, теряя при этом первоначальный смысл и искажая основную идею. Напомним вкратце, в чем эта идея состоит. Если необходимо решить какую-либо сложную, нетривиальную задачу, связанную с использованием совершенно экзотических математических методов, о которых вы имеете слабое представление, или выяснить некий малоизвестный исторический факт, например, уточнить родословную любимой собаки Цезаря, найти и загрузить из Internet последнюю версию драйвера видеокарты от малоизвестного производителя, то дальнейшие действия должны происходить по следующему сценарию (если развитие Internet, конечно, пойдет в направлении, о котором шла речь). Вы активизируете программу-агента на вашем компьютере и в достаточно свободной форме описываете ему интересующую вас задачу. Затем агент связывается с другими сетевыми агентами, пытаясь выяснить, известно ли им что-либо о решении данной задачи. Если находится агент, которому известно решение, то ваш агент осуществляет фильтрацию найденной информации с целью идентификации необходимых решений и отсеивания ненужных данных. Если решение не найдено или является неполным, каждый (или по крайней мере некоторые) из соседних агентов обращаются к соседним агентам, чтобы узнать возможные адреса информационных хранилищ и/или профессиональных "решателей" данной задачи. Процесс продолжается по цепочке точнее, по расширяющейся спирали, охватывающей все большее количество активных агентов. В случае, если за предварительно оговоренный период времени агент, инициировавший запрос, не получает ответа, он сообщает вам о том, что решение вашей задачи современной науке неизвестно [19].

Следует отметить, что приведенный сценарий значительно упрощает содержание реальных процедур, выполняемых агентами: эвристического поиска, интеллектуальных взаимодействий, накопления и обобщения информации, распознавания и классификации. Однако сходство с принципами организации службы имен доменов (DNS-сервиса) действительно имеет место [20], что придает системе агентов гибкость и живучесть, свойственные сервисам Internet.

В последние годы неоднократно появлялись сообщения о создании коммерческих продуктов на базе интеллектуальных агентов. На самом же деле подобные сообщения не слишком соответствуют

действительности. Пока не будут решены некоторые основные проблемы, ожидать появления по-настоящему интеллектуальных агентов не приходится. Среди этих проблем сразу выделим три:

- разработка стандартного языка общения агентов;
- разработка методов эффективной обработки знаний, классификации и распознавания;
- разработка "живого" пользовательского интерфейса.

Главной и весьма специфической проблемой технологии IA [21] является выработка стандарта обмена знаниями и процессе общения агентов. И эта проблема пока остается неразрешенной. Существуют, по крайней мере, две разработки, претендующие на звание стандарта в данной области. KQML Knowlege Query Manipulation Language) и KIF (Knowledge Interchange Format). Что касается KQML, то этот стандарт начал разрабатываться в 1993 году, но до сих пор имеет массу "белых пятен", превращающих проект стандарта в решето. К сожалению, содержание данного документа на сегодняшний день не изменилось (для надежности, проверьте по адресу http://www.cs.umbc.edu/kqm. Идея KQML заключается в создании языка, подобного SQL в реляционному языку для ложных запросов и управления базами знаний. Следует отметить, что разработка стандартного средства обмена знаниями по шаблону SQL DML представляется не вполне обоснованной, поскольку данные и знания имеют все же различную природу и требуют разных подходов к представлению и обработке. Основываясь на предположении о том, что знания являются "просто" метаданными, было проиграно уже немало сражений за создание систем искусственного интеллекта [22]. Вместе с тем, KQML является важным шагом в формировании "общего языка" агентов в сети Internet. Другой стандарт, носящий название "формат обмена знаниями" (KIF: Knowledge Interchange Format) является, на самом деле, форматом построения баз знаний, направленным на создание переносимых структур БЗ. Текущая, третья версия KIF датируется аж 1992 годом. Важно отметить, что KIF является средством построения БЗ, основанных на логических методах представления и обработки знаний, возможности которых весьма ограничены. Но это уже совсем другая история, слишком значительная и драматичная, чтобы стать предметом косвенного обсуждения . Отметим только, что наличие методов эффективной обработки знаний, классификации и распознавания являются необходимым требованием построения системы интеллектуальных агентов. Однако действительно эффективных методов построения промышленной технологии мирового масштаба, к сожалению, пока что нет.

Исследователи в данной области еще не определились окончательно с набором свойств, которым должен обладать стандартный IA [23]. Наиболее радикально настроенные ученые считают, что агент должен обладать синтетической полностой, то есть обладать, вдобавок к средствам коммуникации и обработки знаний, еще и собственным сознанием. Работы в этом направлении ведутся, в частности, в MIT Media Laboratory.

Вообще, идея построения интеллектуального агента как синтетической личности слишком хороша, чтобы использовать ее только в какой-нибудь Web-коммерции или иных бизнес-мероприятиях. Хочется верить, что темные времена для компьютерной науки, отданной в услужение большому бизнесу, рано или поздно завершатся. И тогда мир получит другую Internet – не как очередное средство ускоренного проведения коммерческих сделок и рекламных кампаний, но как мощный ресурс в области научных исследований, образования и культуры. В конце концов, человечество этого достойно.

И в заключение несколько слов о технологии RPSS (Roleplaying Simulation Systems) - системах моделирования ролевых игр. К слову сказать это весьма интересный вариант для разработки компьютерных тренажеров, например, для диспетчеров энергопредприятий. Ставя диспетчера в разные возможные произвожственные ситуации (в игровой форме) и анализировать его действия. При этом можно пользоваться системами поддержки принятия решений, которые необходимо специально разработть для энергопредприятий, чтобы человек мог в критический момент (скажем, аварийная ситуация) правильно на нее среагировать. Все это можно и в дальнейшем должно работать как распределенная сетевая система, что позволит получить выигрыш для всех участников такой системы в связи с интеграцией. Представляется, что в будущем Internet будет выглядеть как система виртуальных миров, населенных синтетическими и естественными личностями. Входя в Internet, можно будет путешествовать по этим мирам, общаться с их обитателями, посещать библиотеки и ставить задачи разнообразным Data WizardaM и Knowledge MagicianaM. Уже сегодня технология RPSS позволяет разглядеть эти будущие черты Internet. Удивительно, но посетить один из таких виртуальных миров можно уже сейчас на сайте фирмы Melhuds. Загрузив небольшой, около двух мегабайт, модуль на языке Java, вы можете стать одним из участников незатейливой демо-версии игрушки в одной из десятка ролей на выбор. И хотя отличить синтетических участников от естественных личностей пока не представляет трудностей, идея игры впечатляет. Речь идет о некоем полусинтетическом сообществе виртуальных интеллектуальных агентов. Фантастика? Сегодня - да. Но завтра ситуация должна измениться.

Просто электронный мир расширяет информационные возможности человечества, переводя их на принципиально новый уровень. Эти качественные изменения должны изменить и реальную сторону жизни, сделав ее шире, богаче, динамичнее. Сейчас повяилась в Internet новая услуга – электронный офис [24]. Его можно арендовать (а есть и бесплатные оффисы), где есть все для работы отдаленных пользователей – средства коммуникации, средства криптографии и обмена информацией. К слову сказать они бесплатны до той поры, пока информацию нет необходимости защищать (то есть пока обмен информация не является конфиденциальной), иначе выход - аренда офиса. Это все может быть весьма полезно для энергетики в самом широком смясле этого слова. Причем, предлагаемый сервис имеет тенденцию только к улучшению, а цены только к снижению. Есть и философский аспект затронутой проблемы: Internet логично укладывается в

концепцию ноосферы, как сферы разума академика В.И.Вернадского. При этом остается проблема качества информации, которая выходит за рамки этой статьи.

Подводя итоги можно сказать, что сегодняшняя Push-технология, завтрашние интеллектуальные агенты и ролевые игры в Internet - все они направлены на решение задач эффективного использования информации и эффективного взаимодействия между людьми посредством глобальных информационных сетей. Хотя многие проблемы ее не решены, перед нами открывается достаточно эффективный путь, в конце которого нас ожидает, может быть, та самая Internet следующего тысячелетия, позволяющая решать насущные задачи электроэнергетики на новом информационном витке.

ЛИТЕРАТУРА

1. Bono P. R., Encarnacao J. L., Hopgood F. R.,Ten Hagen P. GKS-the first graphics standard. Institution of Electrical and Electronic Engineers. Computer Graphics Applications. - 1992, 2, 5, 9-23.

2. Carson G. S. The specification of computer graphics systems. IEEE Computer Graphics Applications, 1993, 3, 6, 27-43.

3. Computer Graphics Standards Committee. American National Standard Functional Specification of the Programmer's Minimal Interference for Graphics. American National Standards Institute. ANSI Document X3H3/82-15rl. 1997.

4. Feuerstein S.T. Advanced Oracle PL/SQL Programming with Packages. – 1999. O'Reilly & Associates, Inc. – 690 p.

5. Encarnacao J. L., Enderle G. Kansy K., Nees G. Schlechtendahl, Weiss J., Wisskirchen . The workstation concepts of GKS and the resulting conceptual differences to the GSPC Core system. Proceedings, SIGGRAPH 1998. In: Computer Graphics, 14, 3, 226-230.

6. Encarnacao J. L., Schuster R., Voge E. (eds). Product data interfaces in CAD/CAM applications. Springer-Verlag. 1998.

7. Enderle G., Grave M., Lillehagen F. (eds) Advances in computer graphics, vol. 1. Springer-Verlag. – 1997.

8. Foley J. Van Dam A. Fundamentals of interactive computer graphics: Addison Wesley. – 1998.

9. Lerman K. Suggested outline for the GKS standard. American National Standards Institute. ANSI Document X3H34/81-1. – 1999.

10. International Organization for Standardization. Information processing systems - computer graphics: Graphical Kernel System (GKS) language bindings. Part 2: Pascal. ISO Document DIS8651/2. Same, Part 3: Ada. ISO Document DIS8651/3. – 1998.

11. Same, Part 4: C. ISO Document TC97/SC21 N669. – 1998.

12. International Organization for Standardization. Information processing systems-computer graphics: Graphical Kernel System (GKS) for three dimensions (GKS-3D) language bindings. Part 1: FORTRAN ISO Document TC97/ SC21 N670. – 1998.

13. International Organization for Standardization. Information processing systems - computer graphics: Programmer's Hierarchial Interactive Graphics System (PHIGS) language bindings, Part 1: FORTRAN. ISO Document TC97/SC21 N667. – 1998.

14. Same, Part 2: Ada. ISO Document TC97/ SC21 N668. – 1997.

15. Same, Part 3: Pascal. ISO Document TC97/SC21/WG2 N496. – 1997.

16. International Organization for Standardization. UK position on extension of CGM. ISO Document TC97/SC21/WG2 N404. – 1998.

17. Newman W. M. Sproull R. F. Principles of interactive computer graphics, (2nd edn), McGraw-Hill. – 1999.

18. Rosenthal D. S., Michener J. C., Pfaff G., Kessener R., Sabin M. The detailed semantics of graphics input devices. Computer Graphics, 1998, 16, 3, 33-38.

19. Артемов А. Internet следующенго тысячелетия: торговая сеть или научный инструмент? – CHIP, 10, 1998.

20. Computer Surveys, 10, 4. Institute of Electrical and Electronic Engineers (special issue) Computer Graphics and Applications. – 1998.

21. Sutcliffe D. C. Attribute handlingin GKS. In: Eurographics 82 (D. S. Greenaway and E. A. Warman, eds), 1998.- pp. 103-110,

22. Duce D. A. Standards for computer graphics. Proceedings, Conference on Electronic Displays. 1995.

23. Smith B. M., Brauner K. M., Kennicot P. R., Liewald M., Wellington 1. Initial Graphics Exchange Specification (IGES) (Version 2.0). National Bureau of Standards, US Department of Commerce. NBS Report 82-2631 (AF). Special Issue (1988). «Graphics standards».

24. Митилино С. Internet как среда коллективной работы / Hot-line. Компьютерное обозрение, 13 (182), 1999.