

Любопытно, что в русском языке слово «менеджер» имеет весьма точный, но основательно подзабытый в эпоху советской России перевод «приказчик». Магическая сила слов! Получил бы популярность клон игры «Монополия», если бы его в конце 1980-х годов назвали «Приказчик»? Теперь сравните пару «менеджер проекта» и «тестировщик» против другой: «приказчик» и «инженер-испытатель»...

Кто не хочет — ищет причины, кто хочет — средства. Ищите возможности сократить путь информации от источника к пользователю и обратно. В конце концов, архитектура служит для эффективной реализации системы, а не освоения бюджета и вовлечения в команду очередных выпускников курсов переквалификации. Подход разработки «по модели», о котором мы ещё поговорим, позволяет генерировать многие слои без участия программистов.

## История нескольких #ifdef

640 килобайт памяти должно  
хватить каждому.

*Билл Гейтс (приписывается), 1980-е годы*

Перелом начала 1990-х годов в России вынудил целые коллективы квалифицированных системно мыслящих людей переходить из специализированных учреждений НИОКР непосредственно на уцелевшие производства, эмигрировать или начинать собственный бизнес. Не мудрено, что в такой ситуации уровень технической культуры в отделе программирования торгово-производственной компьютерной фирмы «Ниеншанц» был достаточным и для софтостроительных проектов. Разумеется, не последняя роль в создании атмосферы и соответствующего интеллектуального уровня принадлежала основателям компании, пришедшим в бизнес из научной среды: Виктору Ярутову, Дмитрию Разгуляеву, Егору Макарову — выпускникам физического факультета ЛГУ<sup>1</sup>.

---

<sup>1</sup> Ленинградский Государственный Университет.

Таким образом, уже в начале—середине 1990-х на предприятии существовала собственная комплексная информационная система Seller 2, выполненная в трёхзвенной архитектуре с технологией если и не совсем «умного», то и не вполне толстого клиента. Хотя разработчики ещё не знали термина *smart client*, это не помешало им аналогичным образом реализовать многие элементы подсистемы представления.

В следующей главе речь пойдёт о разработке тиражной КИС в «Ниеншанце». Разумеется, возникла она не на пустом месте. Чтобы самому не пересказывать предысторию появления нового продукта в виде предшествующих версий, я обратился непосредственно к участникам, своим бывшим коллегам. С небольшими поправками текст публикуется с их согласия.

То, что рассказывают ведущие программисты Дмитрий Цуранов (ДЦ), Сергей Быков (СБ) и Игорь Панышин (ИП), кандидат технических наук, участвовавший в разработке на стадии платформы VAX<sup>1</sup>, вначале было весьма близко к современным понятиям «эволюционный дизайн» и «гибкие методики». Но потом приходит осознание и понимание.

## Начало

**ИП:** Все начиналось на Рижском проспекте (бывший пр. Огородникова), 26 в Институте Аналитического Приборостроения при Академии Наук СССР. Основной нашей деятельностью была разработка программного обеспечения вообще. Нас кидало из стороны в сторону. От разработки графики для терминалов Radiance, отладки кластеров VAX 6320 и VAX station на базе сети DECnet, редакторов, пакетов цифровой обработки сигналов до баз данных Datatrieve, с которыми можно было общаться в командной строке, каких-то игр и просто удовлетворения интереса работой типа «псевдоклавиатурный драйвер». Тогда, на излёте СССР, я постоянно искал дополнительные заработки. Среди моих знакомых был Д. Разгуляев, уже занимавшимся бизнесом, результатом которого стала фирма «Ниеншанц», располагавшаяся в то время около станции метро «Чернышевская».

**ДЦ:** Около 1991 года мы в «Аналитприборе» начали делать систему по заказу фирмы «Ниеншанц». Какую систему? Трудно сказать. Первое техническое задание в виде клочка бумажки с закорючками родилось значительно позже,

---

<sup>1</sup> VAX — 32-разрядная архитектура ЭВМ фирмы DEC, разработка середины 1970-х годов. На базе DEC PDP-11 в СССР была создана линия CM ЭВМ.

а тогда... Тогда мы делали систему, которая работает с базой данных. Не больше и не меньше.

В качестве базы использовалась СУБД VAX RDBMS, а само программирование шло на голом С. Было очевидно, что понадобится прокрутка данных, для чего напрашивались курсоры. Как выяснилось, имеющиеся в VAX курсоры блокировали данные, в результате чего было принято решение отказаться от них вообще, соорудив структуру, называвшуюся «вектор». Вектор — это результат выборки данных, по которой перемещается текущая позиция. Когда вы уходите вверх или вниз, подгружаются новые данные, не одна строчка, а сразу пакет, чтобы обращения к базе были реже. Библиотечка на С, предоставлявшая интерфейс к векторам, получила гордое название CST, или Client Server Tool.

**СБ:** Первоначальная архитектура системы была экзотической — сервер VAX с RDBMS, клиент на ПК. Она была обусловлена тем, что у «Ниеншанца» уже был один VAX. В числе прочих, рассматривался вариант с тонким клиентом: база данных VAX RDBMS, приложение на С или Паскале под VAX/VMS и тонкий клиент на ПК по протоколу X11. Несколько месяцев мы экспериментировали с X Windows.

## #ifdef NWSQL (1991–92 год)

**ДЦ:** Спокойное время в «Аналитприборе», безвозмездно, то есть даром, предоставлявшем нам VAX, кончилось, и мы перешли в фирму «Ниеншанц», где столкнулись с персоналками IBM PC и операционной системой MS-DOS. Впечатление, которое произвела на нас однозадачная MS-DOS после 32-разрядной VAX/VMS с *preemptive multitasking*<sup>1</sup> и защищёнными адресными пространствами, было гнетущим. Но там платили деньги.

В качестве сервера использовался Novell NetWare, а в качестве базы — NetWare SQL, и по этому случаю код библиотеки «векторов» пополнился многочисленными `#ifdef NWSQL`. Так как мы озаботились целостностью данных, то на Watcom C был написан серверный модуль NLM<sup>2</sup>, обеспечивавший

---

<sup>1</sup> Вытесняющая многозадачность — в отличие от корпоративной многозадачности, приложение не способно монополизировать ресурсы операционной системы, поэтому может быть временно приостановлено, а «зависшие» приложения не блокируют остальные.

<sup>2</sup> NLM — NetWare Loadable Module — формат серверного приложения под Novell NetWare. До 4-й версии NetWare работали только в нулевом кольце процессов, то есть при сбое могли вызвать крах всей системы.

механизм пессимистичных блокировок и даже рассылавший сообщения о модификации данных, что проводило к автоматическому обновлению векторов.

Кроме того, у нас появился программист, отвечающий за GUI<sup>1</sup>. Он разрабатывал довольно своеобразный редактор форм. Впрочем, у нас все было своеобразным.

**СБ:** «Ниеншанц» решил, что VAX — это слишком дорого для собственной КИС. Вот тогда возникли Novell и Vtrieve, которые были бесплатным к нему приложением.

Новым программистом GUI был Юрий Дымов по прозвищу «папа», потому что, даже будучи младше нас, в 1992 году он уже был женат, имел дочь. Юра обладал богатым арсеналом приёмов программирования, о котором говорит тот факт, что один раз утечку памяти он пытался исправить сменой компилятора C. Он написал собственный менеджер памяти, то, что современным языком называется *small memory heap*<sup>2</sup>, и заставил конструкторы сторонней графической библиотеки работать через него. Без этого память у нас кончилась бы гораздо раньше...

**ДЦ:** Конечно, программисты такого типа не смущаются вставлять в код уродливые подпорки «чтобы работало». Они обычно плохо работают в длинных проектах, слишком много энтропии вносят в код. Зато если «кровь из носу» надо сделать так, чтобы работало сегодня к 16 часам, — они лучше всех.

**ИП:** В России наступала эра персональных ЭВМ, а я не мог бросить заниматься VAX-ами. Прямо, как чемодан без ручки. Правда, там были общедоступные исходники. Я имею в виду общество DECUS (Digital Equipment Corporation User Society). Поэтому пришлось сделать выбор.

## #ifdef BTSQL (1992-93 год)

**ДЦ:** NetWare SQL был лишь надстройкой к СУБД Vtrieve<sup>3</sup>, встроенной в Novell-овскую серверную операционную систему. Причём эта надстройка выполнялась на клиенте в специальной резидентной программе *brequest*. Она занимала 280 ки-

<sup>1</sup> Graphical User Interface — графический интерфейс пользователя, в данном контексте — терминальное приложение.

<sup>2</sup> Heap — в непосредственном переводе «куча», область динамически распределяемой во время выполнения программы памяти.

<sup>3</sup> Vtrieve не является полноценной СУБД, а относится к так называемым менеджерам записей.

лобайт. Современный человек не поймёт, о чем речь. Он занимал целых 280 из 640 килобайт! Плюс MS-DOS и драйверы, а в оставшихся 300 килобайтах как хочешь, так и крутись... Кроме того, почему-то начальство считало полезным отказаться от «платного» NetWare SQL, будто в России тогда за что-то платили.

В итоге был написан небольшой слой, который находился под CST и транслировал узкое подмножество SQL без соединений (*joins*) в запросы к Vtrieve. Наверное, это худший код, который я написал за свою жизнь, потому что он целиком находился в файле `btsql.c` — пара тысяч строк на чистом C. От обилия глобальных флагов я впал в панику и вводил новые, только чтобы не трогать старый код. Ну а векторы, как и положено, обросли `#ifdef BTSQL`.

Работа с Vtrieve была ещё тем удовольствием, поскольку отсутствовало понятие логического поля. Например, при создании индекса указывалось, что индекс включает байты с 3-го по 9-й, а второй сегмент — с 22-го по 25-й. Список таблиц, полей, смещений к началу полей в записи приходилось вести самостоятельно.

Интересно, что Vtrieve имела уникальный двухверсионный уровень изоляции, который я нигде больше не встречал. Процессы-читатели никогда не блокировались и не считывали «грязные»<sup>1</sup> данные, а если напарывались на них, то брали предыдущую чистую версию. Понятно, что какая-либо целостность этой версии по времени не гарантировалась: каждая таблица хранилась в своем файле. Журналов транзакций не было, «грязные» данные хранились в специальных страницах (*dirty pages*) в той же самой таблице.

**СБ:** Кроме первой пробной установки Novell, за все остальные компания честно платила. Поэтому стоимость лицензий NetWare SQL была серьёзным аргументом в дополнение к его слабому быстрдействию. CST на «голом» Vtrieve работал в разы быстрее.

Кроме системы учёта для «Ниеншанца», на CST в 1992 году была сделана система для Молодёжной Биржи Труда. Как минимум год мы её сопровождали. В том же году сервер CST демонстрировался на выставке в ЛенЭкспо.

## NDL, или Java в миниатюре (1993–94 год)

**ДЦ:** Между тем система, построенная на всем перечисленном, уже активно использовалась в компании и назвалась Seller 1.0. Написана она была, кроме интерфейса, опять-таки на голом C, а из-за ограничения в 640 килобайт пред-

<sup>1</sup> В терминах СУБД «грязными» называются данные неподтверждённой транзакции. Считав их, вы не можете знать, сохранится ли новое значение или вернётся к предыдущему.

ставляла несколько разных исполняемых модулей: `pay.exe` для бухгалтерии, `seller.exe` для продавцов, `store.exe` для склада... Универсальную программу по причине размера собрать было уже нельзя.

Однако ограниченность платформы была ясна, и мы предприняли попытку разработать свой, как говорят сейчас, фреймворк. Да, мы создали в кратчайшие сроки свой язык NDL<sup>1</sup>, компилятор, компоновщик и исполняющую систему, независимую от ОС. Она была переносима без каких-либо серьёзных проблем, в переходный период код исполнялся одновременно и под DOS, и под Windows. В ней была даже реализована бесконечнозначная арифметика с фиксированной точкой, хотя в самом NDL количество знаков ограничивалось 64.

Но Windows будет потом, а пока исполняющая система под DOS научилась грузить NDL-программу в расширенную оперативную память (*extended memory*), освобождая под данные почти всю память основных 640 Кбайт минус исполняющая система, это был грандиозный прорыв. Кроме того, в NDL были исключения (*exceptions*) и куча полезных функций для работы именно с базами данных, так что мы бодро принялись писать новую систему Seller 2 на ней.

**СБ:** Если Seller 1 делался по наитию Д. Разгуляева, уточнявшего каждую неделю техзадание, то Seller 2 мы делали «правильно». Месяца 2–3 мы практически «бездельничали» — то есть придумали и обсудили приличное количество идей, оставив то, что нам казалось лучшим. Нарисовали схему базы данных и даже составили список ключевых функций. И только после этого приступили к реализации.

Сам NDL начинался с интерпретатора. Был придуман мета-код, первые бизнес-функции писались прямо на нем, но через месяц стало ясно, что без нормального языка дальше жить нельзя. Тогда появился NDL, компилятор и компоновщик. За основу синтаксиса был взят Паскаль. Код компилятора генерировался по описанию грамматики конвейером из 2 утилит, `lex` и `yacc`, под FreeBSD, установленной на одном из серверов. Полученный код на C затем компилировался в среде Borland под Windows. Компоновщик NDL собирал проект по описанию в `makefile` с разрешением имён глобальных и локальных ссылок и объектов. При этом он мог оставлять комментарии, которые позволяли исполнять код в режиме отладки с позиционированием на строки исходного кода. Всё было по-взрослому, несмотря на то, что сделано командой из 3 человек менее чем за год.

---

<sup>1</sup> Network Data Language, позднее Nienschanz Development Language.

## Закат Novell

**ДЦ:** Начиналось все хорошо. Вышел новый Vtrieve версии 6, где появились интерактивное резервное копирование (*online backups*) и поиск без индексов. В пятой версии можно было либо читать все данные, либо искать по индексу. А для проверки типа «Сумма=123» необходимо проверять записи на клиенте. Теперь можно было эти проверки перенести на сервер.

Поначалу я этому обрадовался: задал сложный поиск, и сервер думает минуту. Пока не заглянул на заднюю панель компьютера. Лампочки, показывающие активность сетевой карты непрерывно горели! Маленький резидент на клиенте непрерывно посылал запросы типа «Готово? — Ещё нет». Иначе он и не мог, ведь раньше все обращения к Vtrieve были короткими, и ждать не предполагалось...

Но тут вышла Windows 95 под кодовым названием «Чикаго». Novell почему-то не торопилась делать 32-разрядные драйверы и вообще как-то реагировать на новый тренд. В итоге Microsoft сама сделала 32-битные версии драйверов для IPX/SPX, но воспользоваться ими было невозможно, так как пресловутый *brequest* работал в 16-разрядном режиме DOS.

Наконец, я нашёл файл с говорящим именем *breq32.dll* (в «догугловую» эпоху это было делом дней и недель, а не секунд) и... выяснилось, что она представляет собой лишь 32-разрядный интерфейс для обращения к пресловутому *brequest*.

Последний гвоздь в крышку гроба Novell, как платформы разработки, был забит с попытки запустить систему под Windows NT. Все заработало правильно, но раз в 10 медленнее. Так у нас появился следующий *#ifdef*.

**СБ:** У Novell был лучший файловый сервис, который я видел. Нормальное управление правами доступа, *garbage* — мусорная корзина, была реализована на уровне ядра, никаких проблем с восстановлением неосторожно удалённых файлов. Поддержка RAID. Но всё остальное, что делала компания, было ужасно. Их связка Vtrieve + NetWare SQL была тупиком. Они это понимали и родили кентавра OracleWare — NetWare со встроенной СУБД Oracle. Мы были на презентации этого чуда в отеле «Европа».

Сегодня я считаю, что такой альянс был ошибкой, так как Novell базировался на архитектуре x86. Тем, кому был нужен Oracle, такие решения не подходили, они в 1990-х покупали серверы на многопроцессорных RISC-архитектурах, например SPARC, а Novell был продуктом более низкой ценовой категории.

Думаю, они специально не торопились с драйверами под Windows и поддержкой доменов NT в своей службе каталогов, понимая, что Microsoft — их прямой конкурент, поскольку Windows объединила в себе функции сетевой

серверной и настольной ОС. Vtrieve под NT появился в результате выделения продукта в отдельную компанию.

**От автора:** Я снова столкнулся с Novell в 2010 году в рамках небольшого проекта для французской национальной сети телевидения. Корпоративная система безопасности для тысяч компьютеров на разных территориальных площадках была по-прежнему построена на службе каталогов NetWare, хотя и в тесной интеграции с аналогичной службой Microsoft. Новые компьютеры с Windows Vista/7 включались в общую систему. Сама Novell в конце того же 2010 года была куплена малоизвестной компанией Attachmate за целых 2,2 миллиарда долларов и формально прекратила существование. По некоторым сведениям, за Attachmate стояла Microsoft, незадолго до того выложившая 450 миллионов на приобретение у Novell технологий.

## #ifdef Windows

**ДЦ:** Между тем мы стали переходить под Windows. Как уже говорилось, для кода бизнес-логики, написанной на NDL, переделок не потребовалось вовсе. Клиентское приложение было переделано, но понимало описания форм, сделанных ещё под DOS. Конечно, под Windows моноширинные шрифты и формы выглядели довольно уродливо, но тем же страдал и SAP R/3.

Пришлось нам переделывать и систему печати документов под лазерные принтеры, но это отдельная маленькая история. А вот что сильно портило настроение, это старая DOS-подсистема для Vtrieve. Стоило программе обратиться с запросом на сервер, и она уходила в себя. Ещё один камень в огород Novell. Поэтому вскоре у нас появился последний в развитии второй версии #ifdef.

**СБ:** Когда в Seller 2 появился небольшой модуль кадрового учёта, то при запуске система стала поздравлять пользователя с днём рождения. На 8 марта мы рисовали стартовый экран с поздравлениями в стихах. Один год на первое апреля мы перевернули драйвер мыши: двинешь мышшь влево — курсор идёт вправо, мышшь от себя — курсор вниз. В другой раз перевернули экран вверх ногами. Сейчас мне трудно представить, чтобы с утра у всех в конторе, включая директоров, появились перевернутые экраны и программистам за это ничего не было.

Также существовал файл настроек, управлявший цветовой гаммой интерфейса. Кто-то из нас проговорился, и пользователи про него узнали. Такой вахханалии цветов, как в отделе оптовых продаж, я не видел нигде...



## #ifdef MSSQL

**ДЦ:** Система стала полностью 32-разрядной, перепрыгнула на полноценную СУБД. Интересно, что в коде можно было по-прежнему встретить `#ifdef VAXVMS...`<sup>1</sup>

NDL переводил наиболее частые сообщения SQL на русский, чтобы пользователи не пугались. Приложение выдавало «табличку» — окно с сообщением на английском.

**СБ:** «Табличка» — термин одной из работниц склада.

— Валя, почему это табличка? Это же окно сообщения...

— Оно же в рамочке!

**ДЦ:** До боли знакомое всем работающим с SQL Server сообщение *«your process has been chosen as deadlock victim»* было переведено как «Вам дорогу перебежала черная кошка».

**СБ:** Сервер блокировок перестал использоваться после перехода на SQL Server 6.5 и отказа «Ниеншанца» от использования Novell в качестве файлового сервера. Помню, как одна из девочек-продавцов с удивлением узнала, что теперь удалённый файл нельзя восстановить, в корзине сервера его не было. Серверу блокировок стало негде работать, он был реализован в виде NLМ, и его заменили на использование функции `app_lock`. И снова абсолютно прозрачно для бизнес-приложений — просто заменили в NDL реализацию соответствующей мета-команды.

## Постскриптум

Если вы дочитали предысторию до конца, вдумайтесь в цифры. КИС, реализующая основные функции автоматизации деятельности торгово-производственной фирмы среднего размера: от бухгалтерии и складов до сборочного производства и сбыта — была разработана командой из 4–5 человек примерно за полтора года, включая миграцию с предыдущей версии. Система критичная, даже короткий простой оборачивается параличом деятельности фирмы.

Причина столь сжатых сроков? Ясное понимание решаемых прикладных задач, создание соответствующего задаче инструментария, прежде всего, языка

---

<sup>1</sup> Интересно, что если каждую букву аббревиатуры VMS заменить на следующую по алфавиту, то получится WNT, то есть Windows NT. VMS — операционная система компьютеров фирмы DEC, специалисты которой в начале 1990-х годов участвовали в разработке ядра Windows NT.

бизнес-правил высокого уровня, и подтверждение тезиса Брукса о многократно превосходящей производительности хороших программистов по сравнению с остальными.

Последние годы в ходе аудита баз данных я не раз наблюдал, как современные команды в 2–3 раза большей численности, вооружённые умопомрачительными средствами рефакторинга и организованными процедурами гибкой разработки, за год не могли родить работоспособный заказной проект, решающий несколько специфичных для предприятия задач. Сотни тысяч строк кода уходили в мусорную корзину или продолжали поддерживаться с большими трудозатратами, сравнимыми с переделкой.

В другом случае четыре с половиной программиста сумели в короткие сроки создать и в течение многих лет сопровождать КИС для французского туроператора национального (один из крупнейших) и европейского уровня. Это уже потом к ней приделали веб-интерфейс для заказов клиентов и В2В<sup>1</sup>-шлюз с партнёрами.

Есть о чём призадуматься, особенно желающим начать новый проект.

## Ultima-S — КИС из коробки

На дворе стоял 1996 год, падение экономики если уже не прекратилось, то сильно замедлилось, но компания «Ниеншанц» приступила не просто к разработке очередной, третьей версии внутренней системы управления предприятием, а к созданию тиражируемого коробочного продукта. Словно в подтверждение мысли М. Донского о том, что техническая культура — это не производства и знания, а люди, умеющие это делать и применять [11].

Название у нового продукта возникло не сразу. К концу первого полугодия разработки продукта с кодовым названием Seller 3 руководство решило, что для широкого круга потенциальных клиентов название должно быть более «брендовым». Так появилась Ultima-S, где буква «S» перешла по наследству в качестве инициала прежнего имени.

Существует два основных подхода к разработке КИС, условно называемых «от производства» и «от бухгалтерии».

<sup>1</sup> В2В — от англ. business-to-business, шлюз между корпоративными информационными системами.