

# История нашего коллектива глазами А.Н. Терехова

## 1. Трансляторы

Я закончил мат-мех ЛГУ с отличием в 1971 году (первым защитил диплом в первом выпуске кафедры матобеспечения ЭВМ). Моим научным руководителем был доктор физ.-мат. наук Григорий Самуилович Цейтин – сильный и разносторонний ученый. В то время он руководил лабораторией матлингвистики НИИММ ЛГУ, достиг выдающихся успехов в матлогике и теории сложности вычислений, но основным его увлечением было программирование. Именно он увлек группу студентов идеей реализации Алгола 68 – новейшего в ту пору алгоритмического языка высокого уровня [1]. Кроме того, под руководством Г.С. Цейтина в начале 70-х гг. мы реализовали один из первых в стране диалоговых редакторов текста `disco` [2].

В 1974 году я познакомился с Б.А.Бабаяном, под фактическим руководством которого создавалась новая ЭВМ «Эльбрус». Нам поручили разработку интерпретатора Автокода «Эльбрус» на ЕС ЭВМ, ответственным исполнителем был Н.Ф. Фоминых [3,4]. Эта работа получила определенную известность. Поскольку выпуск «Эльбруса» задержался на несколько лет, нашим интерпретатором пользовались десятки коллективов по всему СССР. Некоторые важные заказчики расплачивались с нами фондами на ЕС ЭВМ – в те времена серьезная техника не продавалась свободно, а распределялась по фондам. Платить сотруднику больше полутора окладов было невозможно, поэтому нашим единственным интересом были те самые фонды. Мы говорили, что Коля Фоминых просидит где-то в глуши лесов, устанавливая интерпретатор и обучая как им пользоваться, а с ним расплатятся "борзыми щенками". Таким был наш первый опыт предпринимательства. Разумеется, ЭВМ получал ВЦ ЛГУ, но мы сумели договориться с руководством ВЦ, что "нашими" ЭВМ пользуемся только мы.

С 1972 года заведовать кафедрой матобеспечения ЭВМ в ЛГУ стал чл.-корр. АН СССР Святослав Сергеевич Лавров. Под его руководством в 1975 году мы начали еще более крупную работу по программному обеспечению «Эльбруса» (отладочные средства и несколько десятков пакетов прикладных программ), я в это время уже руководил группой в лаборатории системного программирования ВЦ ЛГУ, наша группа совмещала работы по «Эльбрусу» и Алголу 68.

В 1976 году С.С. Лавров был избран на пост директора Института Теоретической Астрономии АН СССР, его активность в ЛГУ заметно снизилась, в конце концов, он оставил за собой только кафедру матобеспечения ЭВМ, а я был назначен и.о. руководителя большого отдела ВЦ, который был создан специально под работы по «Эльбрусу». Эти работы мы успешно завершили. В дальнейшем наш коллектив от работ по «Эльбрусу» отошел, а С.С. Лавров продолжил со своими студентами, но по другим темам, например, трансляторы с АБВ и Паскаля.

К концу 70х годов работы по транслятору с Алгола 68 для ЕС ЭВМ вышли на заключительную фазу – комплексная отладка на ЕС ЭВМ [5]. Работа приобрела практический, производственный характер, в этот момент Г.С. Цейтин в значительной мере потерял к ней интерес. В течение 2-3х лет мы еще занимались под его руководством изучением и реализацией языков Искусственного Интеллекта и моделированием действий роботов, но большого интереса эти работы в нашем коллективе не вызвали.

В этот период нашим фактическим научным руководителем стал академик Андрей Петрович Ершов. Он возглавил рабочую группу по Алголу 68 ГКНТ СССР, я был назначен его заместителем. Мы собирались 4-5 раз в год в разных городах СССР, обсуждали проблемы реализации Алгола 68 для разных ЭВМ (в том числе, и для

«Эльбруса»). Одновременно разрабатывалось несколько трансляторов с Алгола 68 под руководством Е.Л.Ющенко, А.Н.Маслова, М.Р.Левинсона, А.Н.Терехова и А.П.Ершова.

Хотя полным успехом завершились только работы под руководством А.Н.Маслова (для «Эльбруса») и А.Н.Терехова (для ЕС ЭВМ), все остальные работы также привнесли много новых идей и дали толчок дальнейшим исследованиям. Отдельного упоминания заслуживает русский перевод Официального Сообщения об Алголе 68. В процессе работы под этим переводом в рабочей группе по Алголу 68 были решены и обоснованы многие важные проблемы терминологии, введены в оборот многие понятия программной инженерии.

Осенью 1976 года я в течение трёх месяцев читал лекции в Будапеште и привёз оттуда первые материалы по языку Ада и магнитную ленту с венгерской реализацией языка Пролог. Пролог мы изучили, выполнили несколько экспериментов, но нам декларативный способ не понравился, по крайней мере, на Прологе. Помню, как долго мы возились с задачей сборки деталей пирамидки из разбросанных на столе кружков разного диаметра. Сейчас я понимаю, что декларативный способ может быть с успехом использован, но только для определенного класса задач.

С языком Ада история интереснее. Мы ожидали, что Аду ждёт успех, поскольку это базовый язык Министерства обороны США. Я решил заняться реализацией языка Ада, используя накопленный опыт реализации Алгол 68. Но тут один из сотрудников нашей лаборатории Илья Гиндыш попросил меня отдать эту работу ему. Он на два года старше меня и ему также хотелось попробовать руководить темой. Я с некоторым сожалением согласился. Но уже через полгода руководителем этой темы в нашей лаборатории стал мой молодой выпускник Аркадий Попов. Так я в первый раз понял, что "полки не дают, полки берут". Аркадий Попов решил сделать всё "по науке", а не так, как мы работали над Алголом 68 (мы больше следовали технике быстрого прототипирования). Два года эта группа разрабатывала подробные спецификации, написали два или три толстых тома, но в первые же месяцы реализации натолкнулись на ошибку в проекте, 2 года работы пропали зря. Нет не зря! Мы получили бесценный личный опыт. В конце концов, эта группа завершила реализацию Ады и её окружения на ЕС ЭВМ, а затем и на ПЭВМ. Как и наш транслятор с Алгола 68, реализация языка Ада стала первой в СССР [6].

## **2. Работа с промышленностью**

В 1980 году оборонный отдел Обкома КПСС Ленинграда обратился к ректору ЛГУ В.Б.Алесковскому с просьбой о помощи, а он поручил мне курировать разработки в области ПО для военных заказов. Мне было сказано, что в оборонной промышленности столкнулись с массой проблем при создании ПО систем оборонного назначения, поэтому решено, что ЛГУ в лице нашей лаборатории должен помочь. Моего согласия никто не спрашивал, да в те времена этого и не требовалось. Так мы начали работать с ЛНПО "Красная Заря", "Импульс", "Морфизприбор", "Ленинец", "Аврора", "Гранит" – ведущими предприятиями Ленинграда, работающими в интересах различных родов войск и ведомств. С большинством этих предприятий мы работаем до сих пор.

Первыми из военных организаций к нам обратились сотрудники ЛНПО "Красная Заря" с просьбой помочь в программировании широкого класса задач управления и связи, в частности, создания функционального программного обеспечения (ФПО) телефонных станций, управляемых специализированными ЭВМ (СЭВМ). Несколько лет ушло на изучение предметной области, пробные реализации, решение организационных вопросов.

В это время мы свято верили, что, используя хороший АЯВУ (алгоритмический язык высокого уровня), можно добиться существенного (в разы) повышения

производительности труда. Никаких сомнений, что лучший в мире АЯВУ - это нами любимый Алгол 68, у нас не было.

Небольшой набор базовых операторов, каждый из которых обладает ясной и точно определённой семантикой, при этом достаточно выразителен и нагляден, плюс мощные правила суперпозиции, позволяющие строить сколь угодно сложные конструкции из простых (признаемся, иногда это превращается в недостаток, когда кто-то слишком умный хочет обязательно воспользоваться всей мощностью языка) дают полное право Алголу 68 считаться языком достаточно высокого уровня. Полный видовой контроль периода компиляции обеспечивает высокую защищённость пользователя от ошибок и одновременно обеспечивает потенциальную возможность генерации эффективного объектного кода. Предоставляемые языком возможности описания новых видов (типов) и операций (а также переопределения стандартных операций) в сочетании с возможностью раздельной трансляции и иерархического накопления контекстов дают возможность определять специализированные языки, не выходя за рамки Алгола 68, для которых не надо реализовывать новые трансляторы.

Многие пользователи годами писали программы на кем-то подготовленных для них специализированных языках, не подозревая, что они пишут на Алголе 68 и пользуются одним и тем же транслятором.

Мы начали внедрять Алгол 68 в различные организации. Лучше всего получалось внедрение в военные организации, для которых требования надёжности превалировали над всеми остальными требованиями. Военные для своих задач чаще всего применяли не стандартные ЕС, СМ и персональные ЭВМ, для которых у нас трансляторы с Алгола 68 уже были реализованы, а специализированные ЭВМ (СЭВМ). Пришлось в массовом порядке заняться реализацией кросс-трансляторов.

Очень быстро мы осознали, что все СЭВМ обладают одной общей чертой – они удовлетворяют самым жёстким инженерным требованиям, но программировать для них, а тем более, создавать для них трансляторы, очень тяжело. Видимо, создатели этих СЭВМ не задумывались о проблемах программирования.

### **3. HLL-компьютер «Самсон»**

Мы решили создать свою ЭВМ, следуя принципу «спасение утопающих - дело рук самих утопающих». Эта идея созревала очень долго.

Во-первых, создавая ПО для «Эльбруса», мы прониклись идеей HLL- компьютеров (компьютеров, ориентированных на поддержку основных операторов АЯВУ). Мы любили отечественную ЭВМ «Эльбрус» и не любили «цельносозданные» ЕС ЭВМ И СМ ЭВМ. Но нам не нравилось, что «Эльбрус» очень большой и требует водяного охлаждения, у этой ЭВМ главной задачей было обеспечение сверхвысокого быстродействия. За то, что мы сделали большой комплекс ПО для «Эльбруса», нам дали фонды на эту ЭВМ для мат-меха, который как раз в это время (1979 год) переехал в Петергоф. Мы построили зал площадью 500 квадратных метров, градирню (специальный бассейн для очистки воды), получили первые ящики с оборудованием, но тут университетские инженеры буквально взбунтовались. Они несколько раз съездили на завод–изготовитель, познакомились с правилами эксплуатации и заявили, что в университетских условиях обеспечить все необходимые требования невозможно. С огромным сожалением от идеи установки «Эльбруса» на мат-мехе пришлось отказаться.

Во-вторых, следуя предложенной Н.Виртом идее Р-кода, которую он применял для переноса транслятора с языка Паскаль на разные ЭВМ (то, что у него была и аппаратная реализация в виде ЭВМ Lilith, мы тогда не знали), мы решили разработать полноценную архитектуру виртуальной ЭВМ. Тогда появились первые однобайтовые микропроцессоры

6502 и i8080 и первые персональные ЭВМ. Мы реализовали для них интерпретаторы нашей виртуальной ЭВМ и транслятор с Алгола 68 в её коды (Н.Н. Вояковская, Б.А. Федотов), получив возможность удобной работы в 16-разрядной архитектуре [7]. При замедлении в 4-5 раз мы в 2-3 раза выигрывали в длине объектного кода (за счёт более высокого уровня системы команд), что тогда было очень важно. Чаще всего, первые персональные ЭВМ имели только 64 кб оперативной памяти, но микропроцессоры уже были довольно быстрыми.

Итак, мы решили создать HLL-компьютер, пусть он не будет такой быстрый, как «Эльбрус», но он должен быть маленьким, без водяного охлаждения, но, всё-таки, с приличной производительностью. К этому моменту мы изучили несколько зарубежных HLL-компьютеров, знали мы, разумеется, и архитектуру советского HLL-компьютера «Мир», разработанного под руководством В.М. Глушкова в Киеве. Оказалось, что обычно HLL-компьютер примерно в 2 раза дороже сопоставимого с ним по скорости традиционного компьютера. Хотелось как-то решить эту проблему.

Вначале мы решили сделать специализированный HLL-компьютер. В это время мы уже работали с ЛНПО «Красная Заря», хорошо знали УК1010, СУБК СС, СУБК СМ, «Нева» и другие СЭВМ, ориентированные на телекоммуникационные задачи. Но из идеи специализированного для телекоммуникационных задач HLL-компьютера ничего не вышло. Для установления соединения нужны логические команды, работа с битовыми шкалами, арифметика целых чисел. Для техобслуживания нужна работа со строками, файлы, диалог с оператором, сетевые возможности и т.д. Таким образом, нам нужна именно универсальная ЭВМ, несмотря на то, что она применяется в достаточно узкой предметной области.

Пару лет мы пытались как-то выйти из этой ситуации: универсальные ЭВМ дороги и громоздки, СЭВМ не могут решить задачу полностью. Я знаю АТС, в которой три разных ЭВМ с разными ОС управляют собственно телефонией, техобслуживанием и управлением сетью соответственно. Представляете, сколько нужно офицеров, чтобы обеспечить сопровождение этой АТС?

Решение этой задачи было подсказано нашим опытом разработки трансляторов. Сначала мы заметили, что существенную часть любой ЭВМ составляют схемы контроля, более того, мы видели ситуации, когда ЭВМ не работала именно из-за ошибок в схеме контроля. Отключаешь схему контроля – и всё в порядке! Потом мы заметили, что многие проверки, выполняемые аппаратурой, повторяют аналогичные проверки, которые перед этим делал транслятор с АЯВУ. Поскольку мы разрабатывали и архитектуру ЭВМ, и её системное ПО, мы решили, что подобная избыточность нам не нужна. Мы решили создать ЭВМ, для которой в принципе нельзя составлять программы на ассемблере, тем более, в кодах. Наша технологическая цепочка всегда включает трансляцию с АЯВУ, поэтому то, что проверил транслятор, аппаратура должна принимать "на веру". Зачем нужна проверка на неправильный код операции, неправильный номер регистра или некорректную адресацию, если транслятор такого не генерирует?

Транслятор и операционная система гарантируют, что если на какой-то сегмент памяти есть ссылка (в виде физического адреса) из стека адресов, то этот сегмент не может быть переписан в какое-то другое место в памяти. Поэтому возможна очень эффективная реализация виртуальной памяти. Транслятор может точно подсчитать аппетит каждой процедуры на регистры в стеке, поэтому не нужны аппаратные проверки на переполнение или исчезновение стека. Таких примеров, когда транслятор помогает упростить аппаратуру, нашлось много. Но и хорошо спроектированная аппаратура может облегчить реализацию транслятора. HLL-компьютеры содержат не только простые команды "сложить", "вычесть", "перейти", но и такие сложные, как "вызов процедуры", "цикл", "вырезка элемента массива", что упрощает транслятор.

Очень важна ортогональность системы команд, например, в ЕС ЭВМ нам очень досаждали такие случаи: сложение, вычитание и умножение полуслов есть, а деления полуслов – нет; сложение и вычитание слов требуют одного регистра, а умножение и деление слов – двух и так далее. При проектировании «Самсона» мы стремились избежать таких глупостей, хотя иногда приходилось тратить лишний код операции для очень редко используемой команды, чтобы сохранить ортогональность.

Оказалось, что гарантировать корректную работу аппаратуры мы можем только для статических АЯВУ с полным контролем типов в период компиляции таких, как Алгол 68, Паскаль, Модула 2, Ада, но не таких, как С и PL/I. На этих принципах нам удалось создать компактную ЭВМ, мы дали ей имя "Самсон" [8], но не в честь сильного и легковверного библейского героя, а в честь самого большого фонтана у нас в Петродворце. Мы получили 3 авторских свидетельства (общая организация, совмещение регистров со стекком, организация виртуальной памяти), решили несколько интересных научных задач, привлекли в лабораторию много хороших студентов. Прежде чем был спаян хоть один провод, мы долго экспериментировали на моделях, отражающих потактовую структуру ЭВМ. Многие команды были включены в состав архитектуры именно в результате моделирования.

Реализацией потактного симулятора занималась моя дочь Карина Терехова – тогда студентка младших курсов мат-меха. Мы с ней прогоняли сотни тестов, исследуя статическую и динамическую статистику использования разных команд и определяя, сколько тактов занимает исполнение того или иного теста. После каждого прогона теста мы брали только верхние по частоте 20-25 команд. Штук 5 команд исполнялись миллионы раз, следующие по частоте 10 команд – сотни тысяч раз, еще десятков – десятки тысяч раз, а остальные – сотни или тысячи раз, что для нас было неинтересно.

Верхнюю строку всегда занимала команда «читать из памяти в стек целых» (40%). Нам удалось сделать однобайтовую команду, которая читала на стек одну из первых 15 переменных процедуры. Оказалось, что практически всегда только эта команда и используется, поскольку редко встречаются процедуры с большим числом локальных переменных.

Каждый год на курсе «Архитектура ЭВМ» я рассказывал студентам, как, сидя на колхозном поле, обрубая хвостики морковки, я уговорил Колю Фоминых подумать, как сделать более эффективную реализацию этой команды. Коля полдня отнекивался, а потом все-таки придумал, как вместе 4 тактов сделать 3. Только за счет этого мы повысили производительность «Самсона» на 10%!

Команда «условный переход по ложному условию» занимала 10% всех исполняемых команд (понятно почему – if, while), а команда « $\Leftrightarrow$ » – 8%. Мы их склеили (получив команду «переход по равно») и выиграли 1 байт длины кода и 2 такта исполнения. Таких примеров можно привести множество. Само собой разумеется, из целей ортогональности и облегчения трансляции мы ввели команды «переход» по всем шести типам сравнений, хотя остальные команды использовались не так часто.

Мы объездили несколько заводов в СССР с предложением наладить выпуск нашей машины. Но везде натолкнулись на отказ по той причине, что у нас не было западного прототипа. Такие тогда были времена. Нас выручил завод "Оргтехника" в г. Пловдиве (Болгария), который был побратимом Ленинграда. Все работы были оформлены как подарок к 70-летию советской власти, успешная демонстрация работы новой ЭВМ членам Политбюро БКП и нашим партийным начальникам состоялась 5 ноября 1987 года в 14 часов. В 12 часов "Самсон" еще не работал, мы не спали двое суток, под "горячую руку" попал даже наш ректор В.Б.Алесковский, о чем он со смехом напомнил мне через много лет.

Болгары выпустили 100 экземпляров "Самсона", только после этого нам удалось убедить нашего основного заказчика в СССР – Управление правительственной связи КГБ. Для этого пришлось дополнительно разработать троированную архитектуру с очень высокой надёжностью. Но система команд была точно такая же, поэтому болгарский "Самсон" использовался как инструментальная ЭВМ.

Собственно, идея троированного компьютера (три процессора, три памяти, три входных мажоритара и три выходных) принадлежит не нам. Нам была известна троированная ЭВМ «Микрон», созданная в НИИ АА под руководством академика В.С. Семенихина. Мы использовали такую же структурную схему, но вся реализация была своя.

Очень важной идеей, которую мы реализовали в "Самсоне", было динамическое микропрограммирование. Н.Ф.Фоминих предложил удобную технологию микропрограммирования на Алголе 68 [9], которая в десятки раз упростила эту сложную задачу. Для стандартной системы команд "Самсона" мы занимали не более двух третей микропамяти, оставшуюся треть пользователи занимали для команд, придуманных специально для их приложений. Во многих случаях это позволяло ускорить критичные по времени фрагменты программы в 10-20 раз.

Возможность микропрограммирования дополнительных команд была известна давно, например, все ЕС ЭВМ Ряда 2 этой возможностью обладали. Но никто не пользовался этой возможностью по причине высокой сложности работы. Мы превратили эту работу в деятельность, доступную даже студентам.

Первой реализацией троированного управляющего вычислительного комплекса (УВК) «Самсон» была центральная ЭВМ АМТС «Фобос-К» Управления правительственной связи КГБ [10]. К сожалению, эта АМТС по причинам, весьма далеким от науки и техники, в серию не пошла. Тогда, воспользовавшись связями в Оборонном отделе Обкома КПСС, я договорился с ЛНПО «Импульс» о выпуске этого УВК. Сотрудники «Импульса» (особо надо упомянуть В.П. Самецкого и В.М. Зуева) переработали конструктив и отдельные элементы «Самсона» под свои требования, мы доработали инструментальные средства разработки. В 1992 году министр обороны России Грачев подписал приказ о принятии УВК «Самсон» на вооружение РВСН. Долгие годы это все было большим секретом, но в 1999 году РВСН праздновали свое 40-летие, к этому торжественному моменту была выпущена красочная толстая брошюра на мелованной бумаге. Первые 14 страниц посвящены УВК «Самсон», причем на 10-ой странице есть фраза «Архитектура и ОС УВК «Самсон» разработаны при помощи ГУП «Терком» и ЛГУ». Сначала эта фраза меня покорила, но затем я осознал, что брошюра не содержит грифа, так что секрета больше нет. Теперь в курсе «Архитектура ЭВМ» я рассказываю студентам про HLL-компьютеры на примере УВК «Самсон». Я не думаю, что кто-то из моих выпускников будет на нем программировать, здесь главный интерес – как и почему всё это делалось. Часто перед рассказом какого-то раздела я спрашиваю у студентов, как бы они решили какую-то задачу построения архитектуры и иногда даже получаю ответы.

Сейчас по заказу ФГУП «НИИ автоматики» (г. Москва) мы разрабатываем новый отказоустойчивый вычислительный комплекс (уже прошли испытания макета).

Насколько мы знаем, "Самсон", принятый на вооружение в 1992 году, до сих пор активно используется во многих важных системах.

#### **4. Технология**

Вернемся в самое начало 80х годов в ЛНПО «Красная Заря». Там в течение 15 лет никак не могли завершить работы по самому главному заказу правительственной связи

«Кавказ-1». Меня назначили заместителем Главного конструктора по технологии программирования и поручили помочь в спасении этого «горящего» заказа.

Как уже говорилось, мы, первым делом, попытались внедрить Алгол 68, в частности, реализовали кросс-транслятор в УК1010. Но начинать пришлось совсем не с этого, а с повышения общей культуры программирования разработчиков функционального программного обеспечения (ФПО) для СЭВМ. Дело в том, что нестандартность архитектуры и малая тиражность СЭВМ приводят к отсутствию достаточно развитых операционных систем, трансляторов, средств отладки и других, ставших уже привычными, инструментов программирования. Поэтому мы столкнулись с работой на перфокартах и непосредственно за пультом СЭВМ УК1010 "на тумблерах".

Была сделана попытка воспользоваться известными в то время технологиями, однако, оказалось, что, например, Р-технология не имеет никаких средств настройки на СЭВМ, предлагаемые же в ней графический стиль программирования и программа-организатор с ручным вводом мало помогают в решении задач реального времени; технология РУЗА позволяет автоматизированным образом (но с большими ручными доработками) построить кросс-ассемблер нужной ЭВМ и интерпретатор ее системы команд, а также осуществить некоторую регламентацию работы (например, стандартизовать имена объектов). Р-технология была отвергнута практически сразу (за неимением конкретных программных средств на заданных нам СЭВМ), РУЗА в течение полугода была настроена на одну СЭВМ (УК1010), однако полностью учесть все особенности СЭВМ так и не удалось, кроме того, параметрически настраиваемые кросс-ассемблер и интерпретатор замедляют работу в 5-7 раз.

Мы решили, что из-за таких простых программ, как ассемблер, не стоит "городить огород" и за короткое время реализовали новые кросс-ассемблер и интерпретатор УК1010, действующие на ЕС ЭВМ, которые вместе с текстовым документатором и некоторыми сервисными программами составили основу первой внедренной нами в производство (1984 год) технологической системы, интенсивно используемой сотнями разработчиков ФПО.

В 1986 году «Кавказ-1» был успешно сдан. Разумеется, я не хочу сказать, что именно мы спасли этот заказ, там работали сотни людей, предпринимались жесткие меры, например, сменили Главного конструктора, но, надеюсь, и наш вклад был существенным. Некоторая неудовлетворенность все же осталась. Нам не удалось внедрить наши идеи полностью: во-первых, нас пригласили слишком поздно, во-вторых, мы не хотели влезать в секретные работы, занимались только технологией, а не самим ФПО. Часто программисты «Красной Зари» «забывали гвозди микроскопом», а потом говорили, что это наша технология виновата. Поэтому, когда начался следующий заказ «Фобос-К» того же заказчика, мы взяли на себя технологию программирования (на базе Алгола 68), ФПО и проектирование комплекса вычислительных средств (на базе «Самсона»), соединив, тем самым, три направления наших исследований, ранее развивавшихся независимо (трансляторы, технология, архитектура ЭВМ).

На протяжении всей совместной работы сотрудников ЛГУ и "Красной Зари" боролись две точки зрения на предмет технологии программирования. Сотрудники ЛГУ, в основном, вкладывали в это понятие широкое использование инструментальных средств, а сотрудники "Красной Зари" настаивали на том, что технология - это, прежде всего, набор формальных методик и регламентирующих средств, позволяющих, в частности, на каждом этапе провести экспертизу, архивацию и измерение объема и качества проделанной работы. Такой подход вызывал постоянное раздражение профессиональных программистов.

Только после двух-трёх лет работы в промышленности мы осознали, что нельзя всё сводить к программному инструментарию. Поначалу мы с гневом отказывались от требований начальства детально документировать, кто, что и за какой период написал, но оказалось, что в большом коллективе всегда находятся милые в общении, всеми любимые активисты всевозможных мероприятий, которые вообще ничего не делают по работе. Первую сдачу проекта для правительственной связи я завалил, так как буквально перед самой сдачей кто-то украл одну (!) перфокарту, а эти товарищи всегда начинали с чистой машины и полной перетрансляции. Вредителя так и не нашли, зато я получил хороший урок. Мы быстро реализовали контрольно-учётные программы, архивы с контролируемым доступом, многоуровневые системы сбора версий ПО и тому подобные "шпионские штучки". Так я впервые осознал разницу между "программированием для себя" и "программированием для хозяина", о которых так красочно говорил академик А.П. Ершов.

Проектирование систем реального времени – очень сложная задача, причем трудности носят скорее «человеческий», а не технический характер, то есть проблемы возникают не из-за недостаточной квалификации кого-либо из участников разработки, а из-за взаимного недопонимания. Слишком много людей вовлечено в решение этой задачи – алгоритмисты, специалисты по протоколам, программисты, инженеры-электронщики и другие. Для преодоления этих трудностей Международный Консультационный Комитет по Телеграфии и Телефонии (ныне ИТУ-Т) разработал серию графических языков, в частности, SDL (спецификация Z.100) [11] и MSC (спецификация Z.120) [12].

SDL (Specification and Description Language) диаграмма очень похожа на традиционные блок-схемы, но с несколькими важными расширениями: символ состояния, в котором процесс не занимает процессор, ожидая приема одного или нескольких сигналов; символ приема сигнала и символ отправки сигнала. Эти расширения позволяют удобным образом описывать параллельные процессы.

Первый редактор SDL-диаграмм был нами реализован на ЕС ЭВМ в 1984 году (в это время без всякой ориентации на объектно-ориентированное программирование). Конечно, графическим его можно было назвать лишь с некоторой натяжкой, поскольку и экраны, и устройства печати в то время были только текстовыми. Однако, если забыть про некоторую громоздкость SDL-диаграмм, они выглядели как настоящие.

Мы разработали конвертор из SDL-диаграмм в Алгол 68, систему имитационного моделирования, различные отладчики, средства для снятия и анализа трасс, другие инструменты, которые обеспечили первые практические применения SDL-диаграмм. Первым успешным проектом, разработанным по нашей технологии, стал «Фобос-К».

В начале 90-х годов мы перевели все технологические средства на ПЭВМ, а заодно начали использовать объектно-ориентированный подход к проектированию ПО. Нам удалось найти эффективную реализацию вычислительного процесса для систем реального времени [13]; усилить различные статические проверки, например, если сигнал не упомянут в описании объекта, то его невозможно использовать в SDL диаграммах; разработать средства автоматической генерации данных и так далее.

Под именем RTST (Real Time Software Technology) [14] эта технология просуществовала около 10 лет, с ее помощью были разработано более 10 типов различных телефонных станций и несколько других программно-аппаратных средств, причем 70-80% SDL диаграмм легко переносились с одной платформы на другую. Существенно было облегчено сопровождение ПО, введение в коллектив новых специалистов, переиспользование фрагментов, короче, мы убедились в преимуществах промышленных графических технологий.

Однако постепенно пришло понимание и недостатков RTST. Прежде всего, оказалось, что трудно сразу создать описание классов, да ещё в текстовом виде. Мы уже

начали продумывать графические формы для описания проектируемой системы на этапах раннего моделирования, но тут появился UML [15]. Мы были знакомы с подходами, которые до этого предлагали Буч, Рэмбо и Якобсон по отдельности, но каждый из них охватывал только отдельные фазы жизненного цикла проектирования ПО. Интеграция этих подходов в едином графическом языке UML коренным образом изменила ситуацию, особенно на начальных этапах проектирования. С другой стороны, в 90-х годах UML не имел практически никаких средств для задания детальных алгоритмов. Мы решили использовать наш опыт и объединить подходы UML и SDL в единой технологии [16]. Новая технология получила название REAL – с одной стороны, хотелось подчеркнуть преемственность с RTST и системами реального времени, с другой стороны, мы надеялись, что новая технология окажет реальную помощь создателям ПО.

Постепенно общее развитие технологии REAL разделилось на два направления. Первое связано с традиционной областью применения – системы реального времени. Здесь главной является SDL-диаграмма, в технологию добавляются различные средства снятия и анализа трасс сигналов, имитаторы нагрузки, эффективные способы организации вычислительного процесса.

Второе направление связано с проектированием информационных систем. Лидером этого направления у нас является Александр Иванов [17], который придумал и реализовал средства автоматической генерации различных форм ввода и вывода информации, средства разграничения прав доступа, учета связей между различными атрибутами и т.д. В этом варианте технологии главной является диаграмма классов. По ней автоматически строится описание базы данных на языке DDL (Data Definition Language), программы CRUD (create, read, update, delete, т.е. создать, прочитать, исправить, удалить), API (Application Program Interface – интерфейс программных приложений) для работы с реляционной базой данных как с объектно-ориентированной. По этой же диаграмме генерируются программы работы с визуальными формами, а с помощью диаграммы объектов задаются связи между атрибутами и ограничения прав доступа.

Программа «Студент» [18], автоматизирующая деятельность деканатов и ректората нашего университета (а также многих других университетов) практически полностью автоматически сгенерирована, только очень малый процент подпрограмм был написан вручную на Visual Basic.

## **5. Организационные структуры**

Первые годы мы взаимодействовали с промышленностью традиционным для университета способом – через хозяйственные договора. Но оказалось, что это сильно затрудняет работу: послать сотрудника в командировку к Гензаказчику трудно, выписать пропуск в оборонное предприятие трудно, выплатить деньги за дополнительную работу – тоже трудно. В те годы основным связующим звеном между нами и промышленностью был капитан 1го ранга, доктор технических наук В.П. Морозов. Ещё до работы с нами он прославился тем, что под его руководством были созданы какие-то очень важные программные комплексы для ВМФ, но и он понял, что его «командные» методы имеют свои ограничения, плохо масштабируются. Поэтому В.П. Морозов горячо поддержал идею взаимодействия с Университетом (возможно, он и был инициатором этого). Авторитет у него был огромный – вплоть до ЦК КПСС. Именно В.П. Морозов предложил и за несколько месяцев осуществил идею создания совместного комплексного отдела Минпромсвязи СССР и Министерства образования РСФСР. Когда два министра в 1987 году утвердили положение об этом отделе, дела пошли быстрее, например, наш парк вычислительной техники был мощнее ВЦ ЛГУ, да и зарплаты у нас существенно выросли. Не все в Университете смогли это спокойно вынести, доходило до курьезов. Например, как-то раз меня «пропесочило» партбюро мат-меха по жалобе одного ученого, что, мол,

Терехов обедняет мат-мех, забирая себе лучших студентов, привлекая их возможностью работать на самой современной технике. Выслушав эти претензии, я хмыкнул, что в соответствии с этой жалобой у меня лучшая лаборатория на мат-мехе (лучшая техника, лучшие студенты, самое большое финансирование), поэтому меня надо не порицать, а объявлять благодарность. Пусть этот ученый возьмет на себя пару-тройку оборонных заказов, успешно их выполнит, тогда он сможет конкурировать со мной за лучших студентов. В конце концов, партбюро согласилось с моими доводами, но я хорошо понимал, что играю с огнём – разборки в партбюро тогда часто плохо кончались.

В 1987 году мы с В.П. Морозовым создали НИИ «Звезда», вошедшее в ЛНПО «Красная Заря». В этом НИИ я получил должность «заместитель директора по перспективным научным исследованиям», эту должность мы придумали специально для меня, так как на всех «стандартных» должностях нужно было работать на полную ставку, а я ни за что не хотел уходить из Университета, но был согласен на работу по совместительству. Также поступили и все остальные сотрудники лаборатории системного программирования.

С В.П. Морозовым связано ещё одно важное для меня воспоминание. Со времени образования комплексного отдела у нас сложилась традиция – каждую субботу мы встречались у него дома, где я в течение 5-6 часов рассказывал ему о наших предыдущих исследованиях. В.П. Морозов был великим педантом, все рассказы он протоколировал на миллиметровке в им придуманной форме. В качестве примеров он мне показывал записи десяти- и двадцатилетней давности. Он сразу же составлял план реализации каждой нашей идеи. Поначалу я в это не очень верил, мне казалось, что даже при всех возможностях, его обещания за две недели получить новую мощную ЭВМ, выбить финансирование на НИР по нашей идее, получить заказ на воплощение в жизнь уже проработанной идеи и т.д. – пахивают фантазерством. Но все обещания В.П. Морозова выполнялись. У меня появился чисто спортивный интерес – не пропустить какой-то нашей идеи, разработки, часто даже на довольно далекие от нашей текущей работы темы. Всё шло в ход, пользуясь нашими связями с оборонным отделом, мы легко выходили на руководство разных оборонных предприятий и обычно находили там понимание и сторонников.

Так вот, через 3-4 года наших регулярных субботних встреч я с ужасом обнаружил, что мне нечего В.П. Морозову рассказать. Мы работали как черти, одновременно внедряли кросс-трансляторы, технологии, ЭВМ «Самсон» и различные специализированные устройства, но работы на перспективу как-то незаметно отошли на второй план. Я даже представить себе не мог, что нас можно так быстро иссушить. Этот урок я запомнил на всю жизнь. С тех пор, какая бы важная работа у нас не была, какие бы мы ни использовали финансовые трудности, я всегда трачу часть ресурсов (в последние годы даже личных денег) на поддержание исследований на перспективу.

К началу 90-ых годов ситуация в стране ухудшилась, наши традиционные заказчики перестали финансировать даже вполне прикладные исследования, НИИ «Звезда» вырос до 300 сотрудников, как мы говорили, «забюрократизировался». Чтобы сохранить дееспособное ядро, мы решили в соответствии с модой того времени создать малое государственное предприятие примерно в том же составе, что и совместный комплексный отдел. В.П. Морозов эту идею поддержал, НИИ «Звезда» выступил учредителем. Тогда мы все были абсолютно убеждены, что «смутные времена» закончатся через 2-3 года, поэтому создание МГП мы рассматривали как вынужденную временную меру, даже название ему придумали шуточное – «Терком», т.е. Терехов и его команда.

## 6. Терком и итальянцы

Итак, 14 февраля 1991 года МГП «Терком» было зарегистрировано и начало жить самостоятельной жизнью. Первоначально туда перешли на постоянную работу 30 человек. Я, как всегда, работал генеральным директором по совместительству. Никакого начального финансирования у нас не было. Я читал лекции в Гамбурге и за месяц получил 4000 DM, каждому сотруднику раздал по 100 DM, тогда на эти деньги можно было прожить месяц.

Первый маленький заказ мы получили под работы Петра Лаврова по модификации «Кавказа 1». Нам обещали крупный заказ по «Фобос-К», но его оформление затягивалось. 19-го августа в 6 утра меня разбудили сообщением о путче. Моя жена Галия и Борис Федотов пошли протестовать на Исаакиевскую площадь (до обеда 19 августа там было меньше 100 человек, окруженных несколькими тысячами омоновцев), а я поехал подписывать договор (и подписал!) с ФАПСИ – осколком КГБ. До сих пор моя жена подтрунивает надо мной, вспоминая 19 августа, но 30 голодных ртов, которым ты несколько месяцев не платил зарплату, – это тяжелый груз.

Опыта работы директором у меня не было. Например, я с большим удивлением узнал, что на 3-хлетнюю тему тебе дают 50% денег авансом, а расчет только по окончании темы. Как ты будешь платить зарплату во второй половине срока – это твое дело. Стало ясно, что работать по одному, даже очень большому заказу, нельзя, надо иметь несколько одновременно идущих договоров, чтобы иметь оборотные средства.

Уже тогда у нас появились большие сомнения в стабильности государственных заказов, поэтому мы стали пытаться найти зарубежных заказчиков. Первого такого заказчика нам помогла найти та же «Красная Заря». В 1992 году крупнейшая итальянская телекоммуникационная компания ITALTEL сама захотела наладить сотрудничество с крупнейшим производителем телефонных станций в СССР «Красной Зарей». На встречу с итальянской делегацией Ю.Г. Данилевский пригласил человек 20 начальников отделов и, к счастью, меня. Краснозаревцы работали, в основном, по секретным заказам, до этого им общаться с иностранцами вообще было запрещено, никто не говорил по-английски. Правда, несколько человек имели опыт работы с МККТТ и с французами по телефонной станции МТ20 (Г.С. Истомин, С.Г. Гундарев). Мне было существенно легче, я хорошо говорю по-английски, несколько раз бывал в западных странах, поэтому не был так скован. Когда же я упомянул, что работаю в Университете и руковожу лабораторией системного программирования, итальянцы сразу же захотели съездить к нам в Петергоф. Ю.Г. Данилевский даже обиделся, что я «перетягиваю одеяло на себя», но руководитель итальянской делегации М. Морганти его успокоил, пообещав, что большой кусок работы отдаст «Красной Заре» (и сдержал обещание). Мы же договорились, что будем проектировать устройства и разрабатывать встроенное ПО реального времени. Тут необходимо упомянуть, что в 1991 году я защитил докторскую диссертацию на тему «Технологии программирования встроенных систем реального времени». Этот факт мне сильно помог в переговорах с итальянцами, типичный «рояль в кустах». Итальянцы первыми начали учить нас проектировать кристаллы, наш сотрудник Борис Кривошеин прошел обучение в школе НАТО (как сотрудник ITALTEL), а потом обучал других наших инженеров.

Здесь также надо вспомнить одну поучительную историю. Итальянцы нам заказали разработку SetTopBox – устройства, которое обеспечивает связь с Интернет, используя вместо дисплея домашний телевизор. Мы все сделали в лучшем виде, меня пригласили в Милан для подписания следующего контракта и с гордостью показали огромный ящик, набитый нашими платами. Оказалось, что их целью был вовсе не SetTopBox, а АТМ-коммутатор. Они опасались делиться с нами коммерческими секретами, поэтому работали с нами «втёмную». Я им быстро объяснил, что они сделали нечто дико избыточное,

например, на каждой нашей плате стоит микропроцессор M68000 с обранием, а для целей коммутатора это совершенно не нужно. Они согласились, извинились и тут же заказали нам разработку АТМ-коммутатора «с нуля». Мы и с этим справились. Кажется, это был первый АТМ-коммутатор в Европе, он до сих пор эксплуатируется в сети PEAN. Мы разработали и реализовали и аппаратные средства, и программное обеспечение коммутатора, но, что ещё более важно, реализовали и программные средства управления гетерогенными сетями связи. Насколько я знаю, никто в России до сих пор средств АТМ в полном объеме не разработал.

Наше сотрудничество с ITALTEL развивалось очень успешно, мы занимались передовыми исследованиями, использовали самые новые технологии, публиковали статьи, защищали диссертации. Но всё закончилось самым неожиданным образом. Компания Siemens выкупила 50% акций ITALTEL, а потом оказалось, что немцы просто хотели убить одного из конкурентов. Компания, в которой работало 17 тысяч сотрудников, практически перестала существовать. Одно дело – слушать лекции по марксизму-ленинизму про «гримасы капитализма», но совсем другое – столкнуться с этим в реальной жизни. Мы все были просто в шоке.

Наши коллеги-итальянцы в благодарность за долгое успешное сотрудничество передали нам всё дорогостоящее ПО и оборудование, а также права на интеллектуальную собственность результатов разрабатываемых нами совместно НИР. Даже через 10 лет наши сотрудники защищают диссертации по темам, которые были начаты совместно с итальянцами, например, CoDesign – разработка процессора, оптимизированного на заданный класс задач [Булычев].

Чтобы не заканчивать рассказ про наше сотрудничество с итальянцами на столь грустной ноте, расскажу ещё одну забавную историю. В один из моих приездов в Милан мне устроили экскурсию в «святая-святых» ITALTEL – бизнес-юнит, разрабатывающий основное ПО. Моим экскурсоводом была такая маленькая и веселая итальянка миссис Тутси. Она с гордостью показывала мне их совершенные графические технологии, на мой взгляд, существенно слабее наших. Но, как вежливый гость, я помалкивал. Не удержался я только один раз, когда спросил: «Как же так, у вас описание делается на одном языке (SDL-PR), а поведение описывается на другом (C-SDL)? Если кто-то по ошибке для одной и той же сущности даст противоречивые описания, то кто найдет эту ошибку?». Тут эта маленькая женщина схватила меня за грудки и стала требовать, чтобы я признался, кто выдал этот важный коммерческий секрет, свидетельствующий о потенциальной слабости их ПО. Мне стоило большого труда объяснить, что я не шпион, а профессор, и что я читаю студентам лекции на эту тему. В хорошей технологии все описания должны браться из одного источника, чтобы обеспечить их непротиворечивость.

## **7. Терком и американцы**

Примерно в конце 1992 года через одного бывшего ленинградца (а в то время уже американца) мне в руки попало письмо от американской компании Seer Technologies с приглашением принять участие в проекте по реинжинирингу устаревших программных систем. На самом деле, это письмо не было адресовано мне или нашей компании, они уже сделали несколько попыток как в США, так и в России. Американская бизнес-идея выглядела очень привлекательной: в мире накоплены тонны программ, написанных на Cobol, PL/1, Adabas Natural и других устаревших языках. Эти программы успешно эксплуатируются более 10-20 лет, причем в критически важных областях (оборона, финансы, здравоохранение и т.п.), но их сопровождение с каждым годом становится всё дороже. Авторы программ уже не работают в данной фирме, университеты не готовят специалистов по старым языкам, никто не поддерживает технологии, с помощью которых это ПО было реализовано. Самое страшное, из-за многочисленных изменений, сделанных

в ПО за долгие годы эксплуатации, документация не отражает реальное положение дел. Таким образом, единственным источником реальной информации о программе является её исходный код (текст программы). А поскольку американцы всегда славились своей приверженностью к методу «грубой силы», то приложение объемом 4-5 млн строк, состоящее из модулей по 40-60 тыс. строк на 10-15 разных языках, не является редкостью.

С другой стороны, даже понимание логики работы программы, а, тем более, перевод её на современные платформы, требуют глубокого и детального её анализа (граф потоков данных, граф потоков управления, граф окон взаимодействия с человеком, базами данных и т.д.), а, как известно, числе путей в графе растёт экспоненциально относительно числа узлов в этом графе, т.е. анализ, написанный «в лоб», будет работать очень долго.

В решении этой задачи нам очень помог наш многолетний опыт создания трансляторов и сильная математическая подготовка наших сотрудников. Реинжиниринг дал нам массу совершенно новых задач, по которым было защищено несколько десятков дипломных работ, защищено три кандидатские диссертации [Терехов мл., Эрлих, Мосиенко].

Получившийся в результате продукт Rescue Ware в 2000 и 2001 годах Gartner Group признавала лучшим в области Legacy Understanding и Legacy Transformation<sup>1</sup>. Был период, когда над этим продуктом у нас работало более 100 человек. Мы получили неоценимый опыт разработки программного продукта (а не традиционного для нас сервисного программирования на заказ), настоящей промышленной разработки с выходом на американский рынок. Не обошлось и без скандалов и курьезов.

В 1998 году мы поместили на сайте Теркома краткое описание опыта нашего участия в задаче реинжиниринга с целью найти новых заказчиков на эту тему. Мы не раскрыли никаких тайн, не приводили никаких цифр, оценок производительности, просто хотели сказать, что у нас есть такой опыт.

Американцы устроили настоящий скандал. Они боялись, что продукт, сделанный в России, никто не купит. Пришлось занять жесткую позицию. Я специально летал в Северную Каролину и объяснил им, что мы люди университетские, нам надо публиковаться, защищать диссертации, иначе мы выходим из контракта, пусть они ищут других математиков, которые смогут решить эту задачу. Тогда американцы сменили позицию на 180 градусов, стали везде говорить, что такой продукт смогли сделать только гениальные русские математики, устроили нам интервью по CNN в 19 часов лондонского времени, интервью Associated Press, часовое выступление в клубе журналистов в Вашингтоне и т.д. Часто мы себя чувствовали даже неловко, но уж лучше так, чем вообще нас не упоминать.

Другой случай связан с Андреем Тереховым младшим, который вместе с голландцем С.Verhoef получил премию за лучшую статью по Кобола в 2000 году. В этой статье авторы описывали трудные места семантики Кобола и показывали невозможность «лобовой» трансляции Кобола в современные языки с сохранением семантики. Опять американцы скандалят, что в этой статье, мол, доказывается невозможность Rescue Ware. Мне пришлось снова лететь в Америку и объяснять, что, наоборот, в этой статье показано, сколько трудностей пришлось преодолеть в процессе реализации Rescue Ware и что никто из конкурентов этого сделать не сможет. Вообще, я в Америке был более 15 раз и уверен, что самая частая фраза там: «Я на тебя в суд подам», хотя бизнес там активный и быстрый, не такой, как, скажем, в Германии, где ни одного контракта быстрее, чем за год не подпишешь.

---

<sup>1</sup> Кстати, в 2003 году мы ещё раз были лучшими в этих областях, но уже с другим американским заказчиком.

И еще один поучительный курьез про отношения с заказчиками. Когда американцы впервые приехали к нам в Университет в 1992 году, я принимал их у себя в кабинете, там на полу была большая дыра в линолеуме. Я как-то её не замечал, больше думал, какие у нас персоналки, сеть и т.п. Через 10 лет, когда мы уже стали друзьями, американцы признались мне, что увидев эту дыру, они ровно в два раза уменьшили сумму предлагаемого мне контракта. Ну вот, а я-то думал, что партнеров выбирают по опыту, образованию, знанию английского. Теперь-то я знаю, что в бизнесе мелочей нет.

## **8. Ланит-Терком**

В 1995 году из российского законодательства исчезло понятие «МГП», мы перерегистрировались в государственное унитарное предприятие Терком. В этот момент у нас работало более ста человек, мы стали задумываться, зачем нам это надо. Государственных заказов мы уже не имели, американцы всё время ворчали, что не хотят в нашем лице кормить государство, прибыль мы всю реинвестировали. Сильное давление на меня оказывал А. Терехов (мл.), существенно более бизнес-ориентированный. Лично я к этому времени так ничего и не заработал, а начинать с нуля еще раз не хотелось. Мы стали искать инвестора. Кто только не предлагал нам большие деньги, даже производители водки, но нам казалось это опасным. Были и серьезные предложения, но с условием, что мы станем торговать ПО и компьютерами, а это означало, что надо полностью сменить наш профиль.

В первые годы я не задумывался о таких философских проблемах, как лицо нашего предприятия, его профиль. Мы брались за такие работы, которые нам казались близкими к нашей специальности, которые были нам интересны. Часто оказывалось, что нам доставались такие задачи, за которые никто другой браться попросту не хотел. Постепенно я осознал, что просто программистов в нашем городе, России, да и вообще в мире «пруд пруди», надо обязательно иметь какую-то особенность, которая позволит тебе выделиться, показать потенциальным заказчикам, почему работать надо именно с этим предприятием, а не с многими другими. В те годы все российские компании напирала на дешевизну нашей рабочей силы, но это преимущество быстро улетучилось. Я решил сосредоточиться на наукоемком программировании, до сих пор нашим лозунгом является «smart software solutions». Многие мои молодые сотрудники спорили с этим, говорили, что мы сужаем пространство наших вероятных заказчиков. Сейчас, когда на дворе кризис (начало 2009 года) именно наукоемкие заказчики сохранили свой бизнес и расширяют сотрудничество с нами.

### **Литература**

1. Алгол 68. Методы реализации. Под ред. Г.С. Цейтина. – Л.: Изд-во Ленинградского университета, 1976 г., 224с.
2. С.Н. Баранов, А.Н.Терехов, Г.С. Цейтин «Инструкции к программе DICO», Методические материалы по программному обеспечению ЭВМ. Серия 4. Выпуск 5. Изд. ЛГУ, 1974 г.
3. Н.Ф. Фоминых "Интерпретатор автокода МВК "Эльбрус" для ЕС ЭВМ", диссертация на соискание ученой степени кандидата физико-математических наук, Ленинград, 1987 г.
4. Гиндыш И.Б., Попов А.П., Рухлин А.П., Терехов А.Н., Фоминых Н.Ф., Швецова Г.А. "Интерпретатор АК ЛГУ для ЕС ЭВМ", изд. ЛГУ, 1981 г.
5. Дейкало Г.Ф., Новиков Б.А., Рухлин А.П., Терехов А.Н. "Новые средства программирования для ЕС ЭВМ. Транслятор с языка Алгол 68 и диалоговая система ЛЕС", "Финансы и статистика", М., 1984г.

6. Попов А. П. "Методы реализации языка Ада", диссертация на соискание ученой степени кандидата физико-математических наук, Ленинград, 1987 г.
7. Матиясевич Ю.В., Терехов А.Н., Федотов Б.А. "Унификация программного обеспечения микроЭВМ на базе виртуальной машины", "Автоматика и телемеханика", N5, Москва, 1990 г.
8. Евстюнин М.В., Кожокарь С.К., Терехов А.Н., Уфнаровский В.А. "Как Паскаль и Оберон попадают на "Самсон", Монография по технике программирования. Кишинев, "Штиинца", 1992 г.
9. Н.Ф.Фоминых, Использование стандартного расширяемого алгоритмического языка в микропрограммировании. В сб.: Диалоговые микрокомпьютерные системы. М., МГУ, 1986 г.
10. С.А. Валдаев «Связь, которая не подведет», М., Славянский диалог, 2001 г.
11. CCITT Recommendation Z.100: CCITT Specification and Description Language (SDL) // COM X-R 26, ITU General Secretariat, Geneva, 1992.
12. ITU Recommendation Z.120: Message Sequence Chart. 11/1999.P.138
13. А.Н.Терехов "RTST - технология программирования встроенных систем реального времени", Сб. "Записки семинара кафедры системного программирования "CASE-средства RTST++". Вып.1, СПб, Издательство СПбГУ, 1998 г.
14. Парфенов В.В., Терехов А.Н. "RTST - технология программирования встроенных систем реального времени", Сб. "Системная информатика", Вып.5. Новосибирск, Сибирская издательская фирма РАН, 1997 г.
15. UML Specification version 1.1 (OMG document ad/97-08-11)
16. А.Н.Терехов, К.Ю.Романовский, Дм.В.Кознов, П.С.Долгов, А.Н.Иванов. Real: методология и CASE-средство для разработки систем реального времени и информационных систем//Программирование, 1999 г., №5, с.44-52
17. А.Н.Иванов. Технологическое решение REAL-IT: создание информационных систем на основе визуального моделирования. Системное программирование. Сб. статей/ Под.ред. А.Н.Терехова и Д.Ю.Булычева. – СПб., 2004 г., с.89-100
18. Стригун С.А., Иванов А.Н., Соболев Д.И. Технология REAL для создания информационных систем и ее применение на примере системы «Картотека»//Математические модели и информационные технологии в менеджменте. Выпуск 2./ Под ред. проф. Казанцева А.К. и доц. Должикова В.В. – СПб:Издательство Санкт-Петербургского университета, 2004 г., с. 120-139