

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

СТАНОВЛЕНИЕ НОВОСИБИРСКОЙ ШКОЛЫ
ПРОГРАММИРОВАНИЯ

(мозаика воспоминаний)

Под редакцией

проф. Игоря Васильевича Поттосина

Новосибирск 2001

УДК 007.621.391
ББК 32.81

Становление новосибирской школы программирования (мозаика воспоминаний). — Новосибирск: Ин-т систем информатики им. А. П. Ершова СО РАН, 2001. — 194 с.

Сборник содержит материалы, представляющие становление и развитие новосибирской школы программирования.

Материалы предназначены широкой аудитории как специалистов по информатике, так и представителей гуманитарных наук, интересующихся социальной историей науки.

Подготовка и издание сборника поддержаны грантом РГНФ № 00-03-00277а.



Пока готовился этот сборник пришла печальная весть о кончине **Игоря Васильевича Поттосина**. Он был одним из инициаторов его издания, главным редактором и автором нескольких статей.

Игорь Васильевич Поттосин — профессор, доктор физико-математических наук, один из старейших сотрудников Сибирского отделения Российской академии наук, заведующий кафедрой программирования Новосибирского государственного университета, главный научный сотрудник Института систем информатики им. А.П. Ершова СО РАН, Заслуженный деятель науки Российской Федерации, лауреат премии Совета Министров СССР — скончался 15 декабря 2001 г. после тяжелой непродолжительной болезни.

И. В. Поттосин родился 21 февраля 1933 г. в Куйбышевской области, в 1955 г. окончил спецотделение физического факультета Томского государственного университета по специальности "баллистика". В период с 1955 по 1958 г. он работал в одной из организаций Министерства обороны. В 1958 г. И. В. Поттосин стал сотрудником Сибирского отделения АН СССР, где прошел путь от младшего научного сотрудника Института математики до директора Института систем информатики им. А.П. Ершова СО РАН. Последние годы И.В. Поттосин руководил лабораторией системного программирования Института, возглавлял кафедру программирования НГУ.

И.В. Поттосин был одним из ведущих российских ученых в области информатики. Он внес значительный вклад в становление и развитие

этого направления отечественной науки. За время работы в Сибирском отделении им опубликовано более 120 печатных трудов, в том числе 4 монографии.

Основные научные интересы И.В. Поттосина были сосредоточены в области инструментальной системы поддержки информационных технологий и разработки инструментальных программных средств и их прототипов. Он внес определяющий вклад в разработку методики оптимизирующей трансляции, создал основные принципы реализации и типовые схемы оптимизирующих трансляторов и процессоров, разработал эффективные алгоритмы основных оптимизаций. Эти научные результаты нашли свое воплощение в таких системах, как оптимизирующий транслятор АЛЬФА, система программирования АЛЬФА-6, многоязыковая транслирующая система БЕТА и др.

В последние годы И. В. Поттосин возглавлял проект СОКРАТ по разработке инструментальной системы поддержки программирования встроенных систем, обеспечивающей надежность и эффективность программ. Продолжением этих работ явились исследования в области анализа программ и методологии программирования.

И.В. Поттосин активно участвовал в подготовке специалистов высшей квалификации. Среди его учеников один доктор наук, 11 кандидатов наук, сотни выпускников Новосибирского университета.

Научные и производственные заслуги И.В. Поттосина отмечены премией Совета министров СССР, орденом "Знак Почета", тремя медалями, званием "Заслуженный деятель науки РФ".

Коллектив Института систем информатики им. А.П. Ершова СО РАН искренне и глубоко скорбит о безвременной кончине Игоря Васильевича Поттосина и выражает соболезнования его родным и близким.

ПРЕДИСЛОВИЕ

Предлагаемый читателям сборник содержит материалы по истории новосибирской школы программирования.

Одна из ведущих советских школ в этой области, она ведет свою историю с 1958 года, когда в Институте математики СО АН СССР создается отдел программирования, руководителем которого был с момента основания отдела и до своей кончины Андрей Петрович Ершов — и все публикуемые материалы так или иначе связаны с его именем.

Мы стремились включить в сборник разноплановые материалы — анализ выполненных проектов, историю их создания и использования в российском, советском и международном контекстах, информацию о деятельности отдела, воспоминания о людях и событиях, в которых отразился дух тех 50-х, 60-х, 70-х годов. Часто все это переплетается в одном и том же материале, что нами приветствовалось. И по форме публикации различны — здесь и аналитический обзор, и простой перечень событий и мероприятий, и воспоминания об отдельных событиях, и диалог собеседников: мы не стремились свести все к единому стилю, полагаясь на вкус и желания авторов. Для нас важно, что все публикации — это не взгляд со стороны, они принадлежат людям, принимавшим активное участие в той работе, о которой идет речь. В их основе лежат живые воспоминания, поэтому возможна небольшая абберрация зрения, как это обычно бывает, когда глядишь на давние события.

Первые три публикации дают некоторую общую картину и посвящены основателю новосибирской школы Андрею Петровичу Ершову и работам этой школы периода ее становления. Они в известном смысле являются повторными: первая из них — расширенный вариант доклада, представленного на 4-й международной конференции «Перспективы систем информатики» (Новосибирск, 2001 г.) и опубликованного на английском языке в ее трудах (*Lecture Notes in Computer Science*, Vol. 2244, 2001), две следующие были опубликованы в Трудах Международного симпозиума «Computers in Europe. Past, Present and Future» (Киев, 1998 г.). Учитывая связь этих работ с тематикой сборника и то, что они опубликованы в изданиях, малодоступных широкому кругу читателей, мы сочли возможным открыть ими наш сборник.

Далее следуют две заметки, посвященные университетским годам А. П. Ершова (так сказать, “внутриутробному” периоду развития новосибирской школы) и истории одной из первых прикладных работ отдела программирования.

Последующие шесть работ освещают историю первых проектов отдела в области языков программирования и трансляторов в 60-х годах. Диалог А. А. Берса и А. Ф. Рара рассказывает об участии новосибирцев в международном проекте создания языка Алгол-68. Публикация С. Б. Покровского — это лишь малая часть того, что можно рассказать о столь многолетнем проекте, как многоязыковая транслирующая система БЕТА. Работы Л. В. Городней и Д. Я. Левина освещают новосибирский вклад в реализацию языков, предложенных нашими американскими коллегами и друзьями. А. В. Замулин рассказывает о том, с чего начинались новосибирские работы по информационным системам и базам данных.

Далее Л. В. Городняя рассказывает о деятельности нашего отдела в области школьной информатики — здесь мы были первыми в Советском Союзе. Н. А. Чермных обращает внимание на ту огромную роль, которую А. П. Ершов сыграл в информационном обеспечении научных исследований. Отдел программирования дал жизнь нескольким организациям — историю возникновения одной из них, КБ СП, описывает Е. И. Никольников. Наконец, две последние работы рассказывают об участии новосибирцев во всесоюзной научно-организационной деятельности.

Дополнением к данному сборнику служит отдельно публикуемый препринт В. А. Евстигнеева, в котором даются итоги работ новосибирской школы к концу 70-х.

Сборник заведомо не полон — история ряда других заметных проектов новосибирской школы еще не описана: это предмет следующих публикаций.

Заметим, что история программирования, в отличие, например, от истории вычислительных машин, описана очень слабо: публикаций немного, монографий нет вовсе. Надеемся, что издание этого сборника, пусть в малой мере, исправит это положение.

Написание, подготовка и издание сборника осуществлялись при поддержке Российского гуманитарного научного фонда, грант № 00-03-00277а.

И. В. Поттосин

И. В. Поттосин

А. П. ЕРШОВ — ПИОНЕР И ЛИДЕР ОТЕЧЕСТВЕННОГО ПРОГРАММИРОВАНИЯ

19 апреля 2001 года исполнилось 70 лет со дня рождения Андрея Петровича Ершова. Он был одним из пионеров отечественного программирования и стал его лидером — ученым, влияние которого на становление и развитие отечественного программирования было значительным и определяющим. Этот мемориальный доклад я хотел бы посвятить не столько обзору его научных результатов — я попытался это сделать в другой статье, которая опубликована в сборнике избранных трудов А. П. Ершова — сколько рассказу о нем, как об одном из пионеров программирования и многолетнем лидере в этой области наук. Это будет рассказ о том, кем был Андрей Петрович в отечественном программировании, и чем отечественное программирование ему обязано.

Становление А. П. Ершова как ученого совпало по времени со становлением программирования как науки — в этом специфика его научного пути, отличающая его от его современников, работавших в других областях науки. Проблемы, с которыми он сталкивался, были в большой мере проблемами становления и роста нового научного направления. Ему приходилось быть не только исследователем, но и агитатором, и защитником, и организатором — этого требовала от него возникавшая научная дисциплина. Его научные результаты и организаторская деятельность важны не только сами по себе, но и своей ролью в самоидентификации нового научного направления, в создании основ внутренних исследований этого направления.

Начальной точкой возникновения отечественного программирования следует считать 1950 год, когда появился макет первой советской ЭВМ МЭСМ (и первой ЭВМ в континентальной Европе). Ершов связал свою жизнь с программированием двумя годами позже — когда он стал специализироваться на кафедре вычислительной математики мехмата Московского университета. Он принадлежал к первому в СССР выпуску дипломированных специалистов по программированию — так же, как и его однокашники Э. З. Любимский, В. С. Штаркман, И. Б. Задыхайло, В. В. Луцикович, О. С. Кулагина, Н. Н. Рикко и другие.

Очень интересно вспомнить, как возникло первое поколение программистов. Они уже заведомо не мечтали связать свою жизнь с программированием со школьной скамьи — они, в большинстве своем, рекрутировались из смежных областей — математики, механики, физики, инженерии. Были среди них и педагоги с математическим уклоном. Ершов и его однокашники выглядели среди них элитой — только у них было базовое образование, специализирующее в новой области. Ершов, впрочем, мечтал быть физиком, и только стечение обстоятельств вынудило его прийти учиться на мехмат. Следует заметить, что по словам Е. А. Жоголева, приложившего руку к ориентации Ершова на программирование, именно интерес Ершова к физическому устройству ЭВМ и привел его на кафедру вычислительной математики — единственную кафедру на мехмате, где такое устройство изучалось.

Специфика возникавшей научной дисциплины была таковой, что при своем зарождении она выступала под другой — не чужой, но другой — крышей. Все приходившие в программирование приобретали — независимо от изначальной специализации — марку “математик”. Заметим, что все, работавшие вокруг ЭВМ, делились грубо на два клана. Принадлежавшие к одному клану носили марку “инженер”, принадлежавшие к другому — марку “математик”. Именно последние и были программистами, людьми, создававшими программное обеспечение.

Пребывание под другой крышей, с одной стороны, было естественным — надо было расти под чьей-то кроной, и крона математики была наиболее подходящей: требования к точности и корректности решений, “чистая” интеллектуальность продукта характеризовали как математику, так и программирование. С другой стороны, такие черты программирования, как ценность прагматики, невозможность (по крайней мере, до сих пор) исходить всегда только из доказательных рассуждений, отличали программирование от математики, делали ее в глазах математиков-пуриган некоторой “грязной” ветвью математики. Характерно здесь запомнившееся мне замечание М. И. Каргаполова: «Вот до сих пор существовала, так сказать, теоремная математика, а теперь появилась этакая “бестеоремная” математика». Многие математики, особенно специалисты в вычислительной математике, считали, что единственная роль программистов — это обслуживать запросы вычислительных задач, никакой внутренней проблематики в программировании нет и быть не может. Мнение, что программирование не самостоятельная научная дисциплина, а обслуживающая деятельность, было

достаточно распространено. С другой стороны, правда, бывало такое отношение к программистам, как жрецам нового, непонятного и таинственного бога — компьютера. Если уменьшить эту патетику, разговор программиста с автором задачи напоминал приход сантехника в интеллигентскую квартиру — “мы не знаем и знать не можем эту грубую работу, а вот он знает”.

Противостоять этим настроениям можно было единственным образом — создавать фундамент новой научной дисциплины и строить его модели и методы, свидетельствующие о наличии существенной внутренней проблематики программирования, и тем способствовать его самоидентификации как отдельной научной области.

Ершов как один из пионеров программирования в полной мере испытал трудности становления программирования. Его кандидатская диссертация, законченная в 1959 году и посвященная такой модели программ, как операторные алгоритмы, была защищена только в 1962 году. “Чистые” математики не могли понять той ее ценности, что предлагаемая модель хорошо отражает существенные свойства реальных программ. С постоянной оппозицией Ершову приходилось сталкиваться и при осуществлении его знаменитого Альфа-проекта. Многие не могли понять — как это, коллектив высококвалифицированных программистов вместо того, чтобы писать прикладные, очень полезные программы, занят созданием какого-то транслятора — вещи и бесполезной, и, по своему замыслу создания процессора с языка типа Алгол, получающего программы, близкие по качеству к программам, написанным вручную, невозможной.

Реакция на те внешние трудности, которые возникали на пути становления программирования, могла быть двояка. Можно было “слоупаться” — уйти либо в исследование только чисто математических проблем, связанных с программированием (их, слава Богу, достаточно много), либо в обслуживание вычислительных задач, не претендуя ни на какую самоидентификацию. Можно было обидеться и порвать с математикой, занимаясь лишь конструированием программных систем без всякой попытки привлекать точные методы. Мудрость Ершова была в том, что он не пошел ни по одному из этих путей. Его решением было — одновременное и связанное развитие понятийного фундамента программирования и точных математических моделей, формализующих эти понятия в той мере, в какой возможно. Именно на таком сочетании основывается созданная им новосибирская школа программирования.

Как и для подавляющего большинство пионеров программирования, первые научные результаты Ершова связаны с вычислительными задачами. Его первой статьёй была опубликованная в 1985 году в Докладах Академии наук работа “Об одном методе обращения матриц”, но Ершов не был бы программистом, если бы он сам не написал стандартную программу для БЭСМ, реализующую этот новый, предложенный им метод. Кстати, здесь сказались его начальные математические интересы — прежде чем пойти на кафедру вычислительной математики, он собирался специализироваться по алгебре. Однако с появлением идей, связанных с внутренними проблемами программирования, он, как и многие пионеры программирования, ушел полностью на новую, неизведанную область. Первой такой областью программирования были языки и системы программирования. Он не был самым первым в этой области, но был одним из первых, и сразу выдвинулся. Действительно, целый ряд активных и ищущих программистов с громадным интересом устремились в такую область, которую потом назовут системным программированием. Это были, как и Ершов, будущие лидеры нашего системного программирования — М. Р. Шура-Бура, Э. З. Любимский, С. С. Камынин, Л. Н. Королев, В. М. Курочкин и другие. Ершов был одним из основных разработчиков программирующей программы для БЭСМ — одного из первых отечественных трансляторов. Его идеи были сильными и сразу стали составляющими фундамента концепций языков и методов трансляции. Достаточно вспомнить, что им были предложены (во всяком случае, впервые в отечественном программировании) такая языковая конструкция, как цикл, и такой метод, как функция расстановки (хэш-функция). Им была написана первая в мировой практике монография по трансляции, ставшая широко известной — русское издание 1958 год, английское издание 1959 г., китайское 1960 г.

Начиная со своих ранних работ, Ершов становится одним из ведущих мировых ученых в этой области — области языков программирования и языковых процессоров. Предложенные им идеи и опыт задуманных им и руководимых им проектов — первого оптимизирующего транслятора с языков типа Алгол Альфа, первого кросс-транслятора АЛГИБР, транслятора Альфа-6 для ЭВМ БЭСМ-6, многоязыковой транслирующей системы Бета — стали составной частью современного фундамента трансляции. Методы оптимизации памяти, зачатки методов анализа программ, понятие внутреннего языка как семантического представления программ одного (в Альфа-системах) или широкого класса языков

(в проекте Бета), предназначенного для целей оптимизации программ и/или кросс-компиляции, унифицированная схема трансляции — вот его личный вклад в такой фундамент, основанный на этих системах. Даже такой долговременный и не до конца решивший намеченные проблемы проект, как Бета (некоторыми ехидными коллегами расшифровывающийся как Большая Ершовская Трансляторная Авантюра — что несло оттенок истины: действительно большая и действительно в чем-то авантюрная) внес большой вклад в методику трансляции — и в методах определения внутреннего языка, и во влиянии опыта этого проекта — положительного и отрицательного — на последующие многоязыковые системы.

Он сделал существенный шаг в послеалголовское развитие языков программирования: в Альфа-языке, расширении Алгола 60, был введен ряд черт, характерных для современных языков — многомерные значения, многообразие конструкций циклов, начальные значения и т.п. Предложенный им язык Сигма был примером языкового ядра, расширяемого за счет механизма подстановки.

Широко известными являются его результаты по специализации программ путем смешанных вычислений. Здесь у него были предшественники — Футамура, Турчин — но именно ему принадлежит кристаллизация принципа смешанных вычислений, давшая толчок к появлению большого числа теоретических и практических работ по смешанным и частичным вычислениям. Ученикам Ершова принадлежат работы по практическим и высокоэффективным смешанным вычислениям. Его идеи дали толчок к появлению похожих подходов к специализации программ, в частности, конкретизации. Ершов же сформулировал понятие трансформационной машины как подхода к построению надежных и качественных программ. Все эти идеи Ершова являются основой нового направления создания программ — трансформационного программирования, которое пока еще не нашло большого практического применения, но имеет большое будущее.

Особое место в его работах занимает проект АИСТ. На мой взгляд, значимость и ценность этого проекта были недооценены, а его новаторство было значительным. Здесь Ершов выступал как руководитель создания всей вычислительной системы — и ее архитектуры, и ее программного обеспечения. Начальный этап этого проекта — АИСТ-0 — представлял собой одну из первых отечественных мультипроцессорных систем, снабженную богатым программным обеспечением, реализую-

щим различные режимы обслуживания — от пакетного до разделения времени.

Работы по программному обеспечению в проекте АИСТ происходили практически одновременно с другими работами у нас и за рубежом по созданию развитых ОС, прототипов современных ОС (Малтикс, ОС ИПМ и др.) и были в ряду этих работ достаточно яркими и оригинальными. Архитектура же системы АИСТ была первым серьезным архитектурным проектом, выполненным в Новосибирске, за которым последовали другие оригинальные проекты, выполненные в Институте математики и Вычислительном центре Сибирского отделения. Само программное обеспечение было реализовано по трехуровневой модели: ядро ОС, специализированные ОС и диалоговые системы, прикладное программное обеспечение. Работа была прервана, и последующие этапы не были реализованы из-за принятия, по-видимому, ошибочной национальной программы, о которой Эдсгар Дейкстра как-то сказал: “Наибольшей победой Запада в холодной войне было то, что Советский Союз принял архитектуру ИБМ”.

Первому поколению программистов принадлежит честь создания понятийного аппарата, фундаментального базиса новой научной дисциплины. Они находились в положении первооткрывателей, исследователей мира, который, по выражению Киплинга, “был совсем новенький”. Они черпали понятия, методы из осознания собственного опыта. Ершов с его математической закалкой и опытом руководства рядом принципиально новых проектов, о некоторых из которых говорилось, много сделал для формирования понятийного базиса новой дисциплины. Деталь, но важная: он придавал нужное значение терминологической работе. Ряд фундаментальных терминов русского программистского языка — “информатика”, “программное обеспечение”, “технология программирования” и другие — был привнесен им. Но дело тут, конечно, не только в изобретении названия. За этим стояла и серьезная понятийная и методологическая работа.

По поводу термина “информатика”. Как уже говорилось, Ершов, как и все мы, достаточно долгое время считал себя человеком, работающим в некоторой особой, но математической области, не осознавая до конца специфику того дела, которым занимался. Расстаться с привычной крышей было не так то просто. Однако накопление специфического для новой научной дисциплины багажа рано или поздно потребовало и соответствующего осознания. Но как только этот факт — существование

новой научной дисциплины — был Ершовым осознан, он делает решительный шаг, предлагает новый термин — “информатика”, и в работе 1984 года “О предмете информатики”, опубликованной в Вестнике Академии Наук, очерчивает контуры новой науки, ее смысл и предмет изучения. Он дал термину “информатика” более широкое толкование, чем то, что вкладывается в его традиционный английский эквивалент “Computer Science”, а именно — фундаментальная естественная наука, изучающая процессы передачи и обработки информации. Компьютерная же наука — как она понимается — составляет “рабочее” содержание информатики лишь на сегодняшний день.

Не менее значительную роль сыграл запуск Ершовым в обиход термина “технология программирования”. Программисты из первого поколения привыкли к тому, что создание программ есть высокоинтеллектуальный труд, сродни исследовательскому. В чем-то они были правы, на первых порах так и было. Ершов первым (во всяком случае, среди отечественных программистов) увидел другую ипостась программирования — не исследовательскую, а производственную — как основу новой возникающей промышленности, софтверной промышленности. Очень важно, что Ершов предвидел роль и значимость этой промышленности, необходимость готовить ее кадры, причем разноплановые — архитекторов, руководителей проектов, тестовиков, разработчиков. Иногда он сознательно обострял свою позицию — говорил о “дегероизации трансляции” в том смысле, что создание “хороших” трансляторов с известных языков пусть и сложная, но конструкторская, а не исследовательская задача; ввел термин “фабрика трансляторов”. Такая “промышленная” точка зрения вызвала резкие возражения среди многих, и Ершов достаточно долго и упорно пробивал свою позицию, которая, как оказалось впоследствии, была верной. Термин “технология” в применении к программированию и отражал эту позицию.

Если говорить о фундаменте новой дисциплины — а Ершов с его осознаваемым лидерством понимал всю ответственность формирования такого фундамента и четко видел нужды всего направления — то он подчеркивал две проблемы. Первая — это создание понятийной основы, системы связанных и формализованных понятий, отражающих на хорошем уровне абстракции все, что нужно для процессов создания программ, того, что Ершов назвал лексиконом программирования. Многие в его работах идет в этом направлении, достаточно вспомнить его работы вокруг проекта Бета или работы по смешанным вычислениям. Вто-

рая, связанная с первой, — развитие теории, формальных методов, того, что сейчас четко прослеживается в новых методологиях и технологиях создания программ, в математизации процесса создания программ. В этом направлении он тоже сделал много. О его вкладе в теорию схем программ еще будет говорить, а я хотел бы остановиться на том, что он всегда видел связь между теорией и практикой программирования и умел как увидеть в практических результатах основу для теоретических моделей, так и применить теоретические исследования для практических целей. Еще в Альфа-трансляторе он использовал теорию экономики памяти, созданную им совместно с С. С. Лавровым для построения соответствующих оптимизирующих преобразований. Это было первое практическое применение теории схем программ к обоснованию оптимизирующих преобразований. Эта работа имела далеко идущие последствия. Большое число последующих работ новосибирской школы по созданию моделей программ — линейных, кратных схем, линейных программ, крупноблочных схем, регулярных схем — и по их применению к обоснованию оптимизирующих преобразований выросли из этой работы. Это была, так сказать, наша “Шинель”, из которой мы росли. Еще один пример — наряду с проектом АИСТ, которым он руководил, с его многопроцессной и многопроцессорной обработкой он инициирует работы по формальным моделям параллельных программ.

Ершову принадлежит великолепная и в чем-то уникальная книга — “Введение в теоретическое программирование. Беседы о методе”. Особая значимость этой книги в том, что там ясно показывается, как на основе рассмотрения практических проблем возникают теоретические модели и как эти модели позволяют решать практические задачи. Книга издана в 1977 году, а программистские монографии, мы хорошо знаем, быстро стареют. Но эта монография вне этого — увы — общего правила: в 1990 году вышел ее английский перевод.

Одним из основных проблем возникающего нового направления является воспитание кадров профессионалов и исследователей. Ершов и здесь был и пионером, и лидером. Он много времени уделял становлению образовательной информатики, ее методологии, созданию учебников и учебных систем, пропаганде важности этой области деятельности. Он был признанным главой этой деятельности в нашей стране.

С одной стороны, он занимался проблемой того, как готовить профессиональных программистов — этому был посвящен ряд его работ, а также деятельность на мехмате НГУ, где он создавал коллектив препода-

давателей, обучающих программированию, и сам ставил курс программирования, с другой стороны, он начинал школьную информатику у нас в стране. Здесь его деятельность была многообразна: телевизионный курс лекций, один из первых учебников по информатике для средней школы (он был переведен на многие национальные языки Советского Союза), организация летних школ юных программистов, организация поездки юных программистов в Соединенные Штаты и многое другое. Самое главное, он видел единство и того, и другого: воспитание высококлассных специалистов и повышения информационной культуры общества, начинаемого в средней школе.

Ершов был лидером советского программирования и осознавал это. Лидерство воспринималось им как ответственность — за судьбу и развитие нового направления, за судьбу людей, работающих в этом направлении. Он умел отстаивать значимость программирования и программистов перед внешним миром. У него было прекрасное, далеко не частое свойство: он ценил авторитет знаний, кругозора, ума, а не авторитет должности и положения, и это он относил и к себе. Его высказывания всегда были продуманными, аргументированными, он внимательно следил за собой и никогда не рассчитывал на то, чтобы они воспринимались как неоспоримые истины, неоспоримые потому, что высказываются “большим человеком”.

Я не помню такого случая, чтобы он, зная свой громадный авторитет, вставал в позицию оракула, правого только потому, что имеет высокое положение.

Уже говорилось о том, насколько его научные результаты способствовали становлению, самоидентификации нового направления.

Само определение духа новой специальности, ее этики, ее профессиональной специфики обязано работам и деятельности Ершова. Общественный облик программирования в нашей стране складывался благодаря деятельности таких всесоюзных организаций, как Комиссия по системному программному обеспечению, Совет по кибернетике, Комитет по языкам и системам программирования, которыми руководил Ершов. Его широко известные работы, такие как “Two faces of programming”, “Aesthetics and the human factors of programming”, “Programming, the second literacy”, ясно и выпукло определяли дух и специфику нового вида деятельности.

Он был лидером и в общественной программистской жизни: организатором совещаний и конференций, отечественных и международных,

руководителем всесоюзных комиссий и комитетов, проводником международных связей, создателем информационных потоков (след, оставленный им — Мемориальная библиотека А.П. Ершова при ИСИ СО РАН).

Его безусловное и всеми признаваемое лидерство связано и с его личностными свойствами.

Он обладал истинной стратегией мышления, предвидя будущность только что появившегося явления, прекрасно видел точки роста. Достаточно вспомнить, какие работы он в свое время инициировал — реализацию одного из первых языков спецификации СЕТЛ, работы по технологии создания интеллектуальных систем, работы по теории параллельного программирования, создание системы Школьница — методологически обоснованного инструмента обучению программированию в школе. Уже при появлении персональных компьютеров он понял их громадную будущность, образно назвав первые персональные компьютеры “предками млекопитающих в динозавровом мире ЭВМ”. Помню, как-то мы с Геней Кожухиным, обсуждая основы нашего программистского бытия, пришли к заключению, что если в тактике мы “сами умные”, то в стратегии нам надо доверять мнению Андрея Петровича больше, чем своим собственным — его стратегическая мудрость была для нас очевидной.

Ершов как никто другой из известных мне людей умел радоваться чужим идеям. Он внимательно их выслушивал вне зависимости от того, кто их высказывал, аспирант или профессор, активно их пропагандировал, способствовал авторам в их реализации. Ершов был непревзойденным мастером ведения дискуссий: он умел с ходу схватывать рациональное зерно в любых, даже самых неожиданных высказываниях, выпукло их демонстрировать, конструктивно направлять сам ход дискуссий, не давая им увязнуть в пережевывании одних и тех же положений. Он всегда охотно оказывал помощь и поддержку своим коллегам. Я знаю много людей, для которых поддержка Ершова была весьма существенной. Среди них и эстонец Тыугу, и грузин Цуладзе, и украинец Вельбицкий, и молдаванин Тодорой, и петербуржец Терехов и многие, многие другие. Я уже не говорю о нас, сибиряках.

Итак, Ершов был и одним из пионеров, и лидером отечественного программирования. Если первому он обязан во многом временем, то второму — личным заслугам, величине и значимости своей личности.

Э. З. Любимский, И. В. Поттосин, М. Р. Шура-Бура
ОТ ПРОГРАММИРУЮЩИХ ПРОГРАММ К СИСТЕМАМ
ПРОГРАММИРОВАНИЯ

(российский опыт)

Трансляция — это область, с которой начиналось системное программирование. Первыми программами, принадлежащими системному программному обеспечению, были те или иные трансляторы — ассемблеры и автокоды на Западе, программирующие программы (ПП) у нас.

Возникновение первых идей по трансляции обязано простому соображению: так как программа хранима, как и данные, то она может быть аргументом или результатом некоторых алгоритмов, которые реализуют те или иные аспекты конструирования программ.

Начальные идеи по трансляции были выдвинуты в 1952 году, на самом раннем этапе развития программирования. В этом году Рутисхаузер опубликовал работу [1], в которой изложил идею о том, что по естественной записи арифметического выражения можно построить эквивалентный ей фрагмент машинного кода. Соответствующий алгоритм (многопроходный) был сформулирован. Эта работа дала толчок к возникновению процессоров, в своем входном языке допускающих естественную запись выражений. А. А. Ляпунов в те же годы атаковал проблему с другой стороны. Он заметил, что структура программы включает в себя операторы из небольшого набора типовых операторов и может быть представлена формально в виде строки, соответствующей последовательности операторов программы. На языке операторных схем программа представляется как схема, соответствующая управляющему графу программы, и совокупность спецификаций каждого оператора. Эти идеи А. А. Ляпунов положил в основу курса по программированию, читавшегося им в Московском Университете. Понятие программы как некоторой операторной структуры, взгляд на программу сверху, было революционной идеей, которая практически сразу же легла в основу первых отечественных трансляторов. К сожалению, из-за специфики организаций, в которых работал А. А. Ляпунов, развернутая публикация этого подхода была осуществлена только в 1958 г. [2].

Программирующие программы

Первые отечественные трансляторы носили мнемоническое название программирующих программ. В основе входного языка каждой из программирующих программ лежал общий концептуальный базис, фиксирующий типы операторов и общую идею их спецификации. Унификация языков не ставилась как практическая задача. Как модельный пример может быть приведена следующая схема алгоритма суммирования элементов вектора (в современном виде):

1: A1; 2:A2; 3:A3; 4:A4; 5:F1; 6:P1(3, 7); 7: O1.

со спецификацией операторов:

A1 : $S := 0$

A2 : $i := 1$

A3 : $S := S + a_i$

A4 : $i := i + 1$

F1 : переадресация A3 по i

P1 : если $i \leq n$, то на оператор 3, иначе - на 7

O1 : восстановление A3 в начальном виде.

Типы операторов соответствовали подавляющему большинству решаемых тогда задач, а именно — вычислительным задачам. Выделялись арифметические операторы A i , ведущие вычисление по формулам, логические операторы P i , осуществляющие управление счетом, операторы переадресации F i , позволяющие переходить к следующему значению индекса (и обратные им операторы восстановления), все же неарифметические вычисления объединялись в так называемые нестандартные операторы N i , для которых спецификацией был их машинный код. Специального подязыка описания данных, зачатки которого появились в более поздних ФОРТРАНе и Алголе, не существовало.

Начиналось все с ПП-1. Молодые тогда программисты С. С. Камынин и Э. З. Любимский исследовали реализуемость операторных схем и возможность автоматического построения соответствующих машинных программ. Это было самое начало, и то, что стало очевидным и даже тривиальным сейчас, считалось тогда проблематичным. Законченная в 1954 году программирующая программа ПП-1 [14] являлась одним из первых в мировой практике трансляторов и, по-видимому, имела самый высокий уровень входного языка.

Успешное решение проблемы реализуемости операторных схем как входного языка программирующих программ, осуществленное в ПП-1, дало толчок для двух проектов, разработанных для двух основ-

ных тогда отечественных машин — Стрела и БЭСМ.

Программирующая программа ПП-2 [3], созданная под руководством М. Р. Шура-Буры в 1955 г. для машины Стрела-1, основывалась на ПП-1 как прототипе. Разработчиками были И. Б. Задыхайло, С. С. Камынин, Э. С. Луховицкая, Э. З. Любимский и В. С. Штаркман. В ПП-2 были усовершенствованы алгоритмы трансляции и было уделено заметное внимание оптимизации программ — экономии выражений, оптимальному сочетанию переадресации и восстановления (иначе говоря, наилучшей реализации вычисления индексных выражений), оптимальному отведению памяти для так называемых рабочих ячеек. Это был, по-видимому, первый оптимизирующий транслятор.

ПП для БЭСМ [4], разработанная А. П. Ершовым, Л. Н. Королевым, В. М. Курочкиным, Л. Д. Паниной и В. Д. Поддерюгиным, развивала входной язык программирующих программ. Во-первых, она объединяла схему и спецификацию операторов в одном тексте, во-вторых, был введен первый структурный оператор, а именно — столь существенный оператор цикла (соответствующий современным циклам с параметром).

В этих двух системах была проделана большая работа по созданию фундаментальных алгоритмов трансляции, таких как программирование арифметических выражений, реализация условий переходами и т.п. С использованием идей и алгоритмов этих работ была развернута деятельность по созданию следующего слоя программирующих программ.

Следующие программирующие программы создавались в 1957–58 гг. для различных модификаций ЭВМ Стрела. Надо вспомнить, что серийного производства ЭВМ как такового тогда не существовало, и все производимые экземпляры Стрелы расходились в деталях архитектуры.

ПП-2 послужила идейной основой двух проектов — ПП для Стрелы 4 [5] (разработчики Н. М. Ершова, Е. А. Жоголев, Т. С. Росляков, Н. П. Трифонов и др. при идейном руководстве М. Р. Шура-Буры) и ПП для Стрелы 7 [6] (разработчики А. М. Бухтияров, Л. В. Войтишек, Н. А. Крицкий, А. А. Левина, И. В. Поттосин, Г. Д. Фролов). Важно отметить, что первая из них представляла собой зачаток настоящей системы программирования: помимо собственно транслятора она содержала систему сборки модулей и некоторые средства отладки. Транслятор получал такие фрагменты объектной программы, которые потом назовут модулями, а система сборки создавала программу из оттранслированных модулей и библиотечных программ.

ПП для Стрелы 3 [7] (разработчики Т. М. Великанова, А. П. Ершов, К. В. Ким, В. М. Курочкин, Ю. А. Олейник-Овод, В. Д. Поддерюгин) создавалась под идейным влиянием ПП БЭСМ. В ней был реализован ряд идей, новых тогда, но ставших классическими сейчас — табличный подход к синтаксическому анализу, оптимальное (по числу рабочих переменных) программирование арифметических выражений. А. П. Ершовым был независимо (и практически одновременно с Петерсеном) изобретен метод хеширования [8], который применялся к экономии арифметических выражений.

Итак, как и обычно при становлении нового направления, одновременно с созданием первых российских трансляторов — программирующих программ — шел процесс кристаллизации и оформления базовых идей и подходов и одновременно процесс создания коллективов квалифицированных исследователей, сказавших свое слово и на последующих этапах развития этого направления. Как некоторый итог этого начального этапа представляется вышедшая в 1958 г. первая в мировой науке монография А. П. Ершова [9], посвященная проблемам трансляции. Монография практически сразу была переведена и издана за рубежом [10].

Нужно заметить, что с трансляции начиналось не только системное, но и теоретическое программирование. Именно работы по входным языкам (операторным схемам) и трансляторам (программирующим программам) послужили толчком к созданию первой математической модели программ — схемам Янова [11]. Начальным толчком к появлению такой известной модели, как стандартные схемы стала работа А. П. Ершова, одного из создателей первых трансляторов по операторным алгоритмам [12].

Алголовские трансляторы

Советские работы 1953–58 гг. развивались независимо от мирового опыта. Практически параллельно с отечественными работами появились серьезные труды по трансляции и в США. Сопоставимый по уровню входного языка известный транслятор с ФОРТРАНа был описан в статье Дж. Бэкуса [12], появившейся в 1957 г. Реальное привлечение мирового опыта к отечественным разработкам по трансляторам началось с возникновением Алгола, точнее — Алгола-58.

Параллельно с работами по программирующим программам развивался и подход к автоматизации программирования с помощью библиотек стандартных программ. Е. А. Жоголевым была разработана стан-

дартная составляющая программа — ССП [15], которая осуществляла статическую загрузку и связывание стандартных программ из достаточно обширной библиотеки, был предложен достаточно амбициозный проект организации библиотеки и подключения подпрограмм [16]. По-видимому, кульминацией этого подхода было создание М. Р. Шура-Бурой интерпретирующей системы (ИС) ИС-2 [17], которая использовалась повсеместно на новых машинах М-20 и учитывалась всеми трансляторами нового поколения. В ИС-2 был весьма эффективно реализован способ динамического подключения библиотечных подпрограмм. ИС-2 реализовывала некоторые функции будущих операционных систем, осуществляя динамическое связывание, подкачку и смену используемых подпрограмм, причем все это делалось с небольшими накладными расходами и весьма скромными запросами на память. Высокая эффективность ИС-2 и хорошо продуманный интерфейс с основной программой сделали ее (как и положено операционной системе) неотъемлемой частью комплекта поставки ЭВМ.

Существенно новым шагом в развитии направления трансляции стало создание совокупности алголовских трансляторов для появившейся массовой (и уже серийно производившейся) машины М-20. Хотя язык операторных схем имел уже достаточно высокий уровень, вполне сопоставимый с уровнем тогдашней версии ФОРТРАНа, Алгол-60 был качественно новым языком, ставившим ряд новых важных задач. Вместе с тем в стране уже был накоплен багаж базовых идей и созданы коллективы высокопрофессиональных специалистов в области трансляции, что привело к возникновению оригинальных и широко использовавшихся систем программирования.

При всех своих достоинствах программирующие программы не имели достаточно широкого практического использования: даже при их наличии почти во всем множестве архитектур, существовавшем в стране, доля программирования непосредственно в машинном языке была преобладающей. Переход к современному стилю программирования на языках программирования высокого уровня был осуществлен благодаря созданию первых алголовских трансляторов. Произошло так, что первая тройка отечественных алголовских трансляторов, созданная для наиболее массовой тогда ЭВМ М-20, различалась по своим потребительским возможностям так, что своей совокупностью они удовлетворяли практически всем категориям пользователей.

Работы над этими трансляторами (ТА-2 [18], ТА-1 [19] и Альфа [20])

начались практически сразу после публикации завершеного международной рабочей группой описания языка. Хотя, по-видимому, первым алголовским транслятором стал транслятор Айронса [21], но и ТА-1, и ТА-2 были завершены в 1963г., т.е. практически одновременно с другими известными западными работами — П. Наура [22] и Э. Дейкстры [23]. Транслятор Альфа ввиду объемности реализации был завершен годом позже.

Транслятор ТА-2 (разработчики И. Х. Зусман, С. С. Камынин, Д. А. Корягин, А. С. Луховицкая, В. В. Луцикович, Э. З. Любимский, В. Б. Мартынюк, Г. М. Олейник-Овод, В. И. Собельман под руководством М. Р. Шура-Буры) встретил вызов Алгола-60 грудью, не вводя практически ограничений ни на язык, ни на размеры входных программ. В трансляторе был разработан оригинальный алгоритм программирования процедур (понимаемых как потенциально рекурсивные), мощные механизмы управления математической памятью как единства оперативной и внешней, систематически применялся метод таблично-управляемой генерации кода. В мае 1963 г. первым из всех трех этот транслятор уже демонстрировался на международной конференции по автоматизации программирования в Киеве.

Транслятор ТА-1 (разработчики В. Н. Попов, В. А. Степанов, А. Г. Стишева, И. А. Травникова под руководством С. С. Лаврова) основывался на быстрой и простой схеме трансляции (в частности, был разработан стековый подход к программированию выражений). Для достижения такой схемы авторы отказались от ряда средств Алгола-60: рекурсивности процедур, наличия статических (собственных — own) объектов в блоках, от возможности введения параметров процедур без их типизации и пр. Вместе с тем, что существенно для современных систем, особое внимание было уделено простоте и удобству эксплуатации.

В Альфа-трансляторе (разработчики Г. И. Бабецкий, М. М. Бежанова, Ю. М. Волошин, Б. А. Загацкий, Л. Л. Змиевская, Г. И. Кожухин, С. К. Кожухина, Ю. И. Михалевич, Р. Д. Мишкович, И. В. Поттосин, Л. К. Трохан под руководством А. П. Ершова) особое внимание уделялось эффективности получаемых программ. Большой набор оптимизирующих возможностей (квазилокальная экономия выражений, оптимальное и весьма изощренное программирование процедур, циклов и индексных вычислений, глобальная экономия памяти и пр.) давал возможность получать объектные программы, мало отличающиеся по времени исполнения от созданных вручную программ. Платить за это при-

ходилось большим объемом транслятора, большим числом проходов (24 прохода), отказом от некоторых средств Алгола, оптимальная реализация которых не была найдена (в частности, как и в ТА-1, рекурсивность была запрещена). Вместе с тем входной язык являлся расширением Алгола (точнее, его подмножества), включавшим ряд таких типичных для современных языков средств, как операции над многомерными значениями, комплексный тип, начальные значения переменных и пр. В целом в Альфа-трансляторе начала складываться современная методология оптимизирующей трансляции с такими ее понятиями, как внутренний язык, многопроходная реализация оптимизирующих преобразований, потоковый анализ.

В целом, как уже говорилось, эта совокупность трансляторов покрывала разнообразные запросы пользователей. Для большого потока сравнительно небольших программ без особых требований к эффективности подходил ТА-1, программистам, которые хотели воспользоваться всеми возможностями Алгола, в том числе и возможностями транслировать большие задачи, был пригоден ТА-2, а для тех, кому надо было программировать задачи с большим счетом и, значит, обязательно иметь эффективный код, существовал Альфа-транслятор.

В отличие от программирующих программ все алголовские системы были уже настоящими системами программирования, близкими к современным: они включали в себя не только собственно транслятор, но также отладочные и некоторые редактирующие средства. В ТА-1 и Альфа-систему входили отладочные редакторы, которые по заданию на отладку создавали отладочную версию программ. Говоря по современному, платформой для всех систем была ИС-2, все системы включали ее средства в свой входной язык и строили программы, ориентированные на ИС-2.

Языки системного программирования

Иницированное созданием трансляторов появление системного программирования поставило задачу создания адекватных языков программирования. Все языки, появившиеся к началу 60-х годов — Фортран, Кобол, Алгол-60 и др., — не учитывали этой области программистской деятельности, и алголовские системы, будучи достаточно большими программными системами, писались еще в машинных кодах вручную. Адекватные языки нужны были не только для трансляторов, но и для всей возникающей области системного программирования: нача-

ли появляться и другие языковые процессоры, и первые операционные системы, и информационные системы — все то, что потом назовут базовым программным обеспечением.

В связи с этим и у нас, и на Западе начали появляться специальные языки, предназначенные для системного программирования. Характерной чертой для первого поколения этих языков (связанной с необходимостью хорошо учитывать архитектуру и машинное представление данных) была машинная ориентированность. Разрабатывались эти языки, как правило, в коллективах, имевших большой опыт в создании системных программ — а таковыми в тот период были системы программирования. Отечественные языки Алмо [24], Эпсилон [25], Сигма [26], которые были одними из первых в мире языков системного программирования, создавались поэтому в коллективах трансляторщиков, только что завершивших большие программные проекты и почувствовавших, каково сапожнику обходиться без сапог.

Язык Алмо имел в своей основе некоторую абстрактную машину, отражавшую особенность существовавшего тогда класса машин, и в этом отношении Алмо-подход предвосхищал появившиеся позже Р-код, М-код и прочие подобные подходы. Помимо того, что этот язык был языком реализации для системного программирования, он предназначался быть языком-посредником при трансляции с различных языков. Идея была в том, чтобы заменить трансляцию с m входных языков в n машинных языков трансляцией “из m в один” и “из одного в n ”. Важно заметить, что в языке существовали средства определения многопроцессности (типа `fork` и `joint`). Машинная ориентированность явно прослеживалась в языке — регистровые объекты, постфиксная запись выражений, оперирование с битами машинных слов и т.п. Были созданы реализации языка для основных отечественных машин того времени (М-20, БЭСМ-6, Минск 2, Урал 11) и трансляторы с Алгола-60 и ФОРТРАНа в Алмо, причем все трансляторы также были написаны на Алмо и “раскручены” на всех этих машинах.

Язык Эпсилон трактовал машинную ориентированность иначе. В этом отношении его идеология совпадала с одновременно появившимся языком Н. Вирта ПЛ360, ставшим прототипом большого числа языков системного программирования того времени: средства высокого уровня необходимы для естественного (для программиста) одевания конструкций машинного языка. Семантика каждой конструкции языка определялась сопоставленным ей образом — последовательностью операций

машинного языка. При этом, конечно, рекомендовалось, чтобы образы для разных машинных языков были бы в определенной мере аналогичны, однако переносимость не обеспечивалась, за ней надо было следить. Типы данных языка тоже следовали за назначением языка и его машинной ориентированностью — допускались целые (но не вещественные), классы двоичных значений, слоги машинных слов или их последовательности. Эпсилон тоже был реализован для большого числа архитектур — М-20, БЭСМ-6, Минск-2. Существенно, что при создании многопроцессорной архитектуры АИСТ-0, Эпсилон играл такую же роль, как впоследствии Си для Юникса или Эль-76 для Эльбруса — был базовым системным языком, на котором писалось все многочисленное системное обеспечение, от ядра ОС до системных программ, ориентированных на пользователя.

Язык Сигма обладал двумя важными и в то время новыми особенностями. Синтаксически ограничиваемый макросами, он, как и более поздний Форт, допускал генеральную линию создания программ — подстановку описанных макросов. Язык содержал средства формального описания конкретной архитектуры: была разработана система параметров, в терминах которой фиксировалось представление языка для конкретной ЭВМ. Таким образом, общая Сигма-программа вместе с описанием архитектуры ЭВМ транслировалась на данную ЭВМ. Параметрами архитектуры были длина слова, представление значений типов в машинном слове и т.п., так и правила заполнения шаблонов машинных команд.

Заключение

Таким образом, вспоминая славные дни становления у нас в стране трансляции как области исследований и разработок, авторы, как кажется, сумели показать, что российские коллективы всегда находились на высоком научном уровне. Подходы и системы, о которых здесь рассказывалось, были выбраны в качестве той линии, к которой авторы имели непосредственное отношение. Часть российских работ, лежащая вне этой линии, здесь не рассматривалась — более широкое рассмотрение всех работ тех лет можно найти в обзоре [28, 29], а языки системного программирования тех и более поздних лет рассматриваются более подробно в [30].

Цитируемые источники

1. **Rutishauser H.** Automatische Rechenplanfertigung bei programmgesteuerten Rechenmaschinen // *Miff. Inst. fur Angew. Math. dec ETH Zurich.* — N 3. — 1952.
2. **Лягунов А. А.** О логических схемах программ // *Проблемы кибернетики.* — Вып. 1. — М.: Физматгиз, 1958. — С. 46–74.
3. **Камынин С. С., Любимский Э. З., Шура-Бура М. Р.** Об автоматизации программирования с помощью программирующей программы // *Проблемы кибернетики.* — Вып. 1. — М.: Физматгиз, 1958. — С. 135–171.
4. **Ершов А. П.** Программирующая программа для БЭСМ АН СССР // *Тр. конф. “Пути развития советского математического машиностроения и приборостроения”.* — ч.3. — М.:ВИНИТИ, 1956. — С. 18–29.
5. **Система автоматизации программирования** // Под ред. Н. П. Трифонова и М. Р. Шуры-Буры. — М.: Физматгиз, 1961. — 187 с.
6. **Бухтияров А. М., Войтишек Л. В., Левина А. А. и др.** Программирующая программа ПП-С // *Сб. научных трудов № 1.* — МО СССР. — М., 1959.
7. **Великанова Т. В., Ершов А. П., Ким К. В. и др.** Программирующая программа для машины “Стрела” // *Тез. докл. сов. по вычисл. матем. и прим. средств выч. техники.* — Изд-во АН АзССР. — Баку, 1958.
8. **Ершов А. П.** О программировании арифметических операторов // *Докл. АН СССР.* — 1958. — Т. 118, № 3. — С. 427–430.
9. **Ершов А. П.** Программирующая программа для быстродействующей электронной счетной машины // *Изд-во АН СССР.* — М., 1958. — 116 с.
10. **Ershov A. P.** Programming program for the BESM computer / Pergamon Press. — London, 1959. — 158 p.
11. **Янов Ю. И.** О логических схемах алгоритмов // *Проблемы кибернетики.* — Вып. 1. — М.:Физматгиз, 1958. — С.75-127.
12. **Ершов А. П.** Об операторных алгорифмах // *Докл. АН СССР.* — 1958. — Т. 122, № 6. — С. 67–970.
13. **Backus J. W.** The FORTRAN automatic coding system / WJCC. — 1957. — P. 188–198.
14. **Камынин С. С., Любимский Э. З.** Автоматизация программирования // *Тр. конф. “Пути развития советского математического машиностроения и приборостроения”.* — Ч. 3. — М., 1956. — С. 9–17.
15. **Жоголев Е. А., Росляков Г. С., Трифонов Н. П., Шура-Бура М. Р.** Система стандартных подпрограмм // *М.:Физматгиз.* — 1958. — 231 с.
16. **Кожухин Г. И., Нагорный Н. М., Поттосин И. В.** Принцип организации и использования библиотеки программ // *Вычислительная математика.* — Изд-во АН СССР. — М., 1961.

17. **Шура-Бура М. Р.** Система интерпретации ИС-2 // Библиотека стандартных программ. — М.:Изд-во ЦБТИ, 1961.
18. **Шура-Бура М. Р., Любимский Э. З.** Транслятор АЛГОЛ 60 // ЖВМиМФ. — 1964. — Т. 4, № 1. — С. 96–112.
19. **Попов В. Р., Степанов В. А., Стишева А. Г., Травникова Н. А.** Программирующая программа // ЖВМиМФ. — 1964. — Т. 4, № 1. — С. 78–95.
20. **Бабецкий Г. И., Бежанова М. М., Волошин Ю. М. и др.** Система автоматизации программирования АЛЬФА // ЖВМиМФ. — 1965. — Т. 5, № 2. — С. 317–325.
21. **Irons E. T.** A syntax directed compiler for ALGOL 60 // SACM. — Jan. 1961. — №4. — P. 51–55.
22. **Naur P.** The design of the GIER ALGOL compiler // BIT. — 1963. — № 3. — P. 124–140.
23. **Dijkstra E. W.** An ALGOL 60 translator for the X1 // Ann.R. — 1963. — № 3. — P. 329–356.
24. **Камынин С. С., Любимский Э. З.** Алгоритмический машинно-ориентированный язык АЛМО // Алгоритмы и алгоритмические языки. — Изд-во ВЦ АН СССР. — 1967. — Вып.1.
25. **Поттосин И. В., Рар А. Ф., Катков В. Л.** ЭПСИЛОН — система автоматизации программирования для задач символьной обработки // Тр. 1-ой Всесоюзной конф. по программированию. — Киев, 1968.
26. **Ershov A. P., Rar A. F.** SYGMA, a symbolic generator and macroassembler // Symbol manipulation languages and technique, North-Holland Publ. Co., 1968.
27. **Wirth N.** PL 360, Programming Language for the 360 Computers // JACM. — 1968. — Vol. 15, № 1.
28. **Ершов А. П., Шура-Бура М. Р.** Становление программирования в СССР // Кибернетика. — 1976. — № 6. — С. 141–160.
29. **Erschov A. P., Shura-Bura M. R.** The Early Development of Programming in the USSR // A History of Computing in the Twentieth Century. — Academic Press, 1980. — P. 137–196.
30. **Поттосин И. В.** Языки реализации для системного программирования // Новосибирск, 1979. — 24 с. — (Препринт/Сиб. Отд-ние АН СССР. ВЦ; № 179)

И. В. Поттосин

А. П. ЕРШОВ И СТАНОВЛЕНИЕ НОВОСИБИРСКОЙ ШКОЛЫ ПРОГРАММИРОВАНИЯ

Основоположник

Становление ведущих научных школ по информатике у нас в стране практически всегда было связано с именами ярких личностей. Если у истоков киевской школы стоял В. М. Глушков, а у истоков ИПМовской школы — А. А. Ляпунов и М. Р. Шура-Бура, то у истоков новосибирской школы стоял их более молодой соратник А. П. Ершов, ученик А. А. Ляпунова.

Сибирское отделение АН СССР было создано в 1957 году, а год спустя, осенью 1958 года, было создано ядро будущей новосибирской школы программирования — отдел программирования Института математики СО АН СССР. Фактическим, а потом и формальным его руководителем стал А. П. Ершов. Несмотря на свои молодые годы, А. П. Ершов к этому времени уже стал известным специалистом в области трансляции и одним из лидеров этого направления в стране — он был руководителем двух проектов программирующих программ, ПП БЭСМ и ППС, автором первой монографии по трансляции. Набранный им коллектив был тоже молодежным, что, впрочем, не удивительно: подавляющее большинство всех программистских коллективов было молодо, новая наука бурно втягивала в себя молодежь. Первыми сотрудниками отдела программирования стали выпускники или недавние выпускники университетов Москвы, Ленинграда, Томска, Риги, Саратова.

Проект Альфа

Первым большим проектом, с которого начиналась новосибирская школа, был трансляторный проект “Альфа”. В отличие от предыдущих проектов программирующих программ проект Альфа был хорошо, практически по современному специфицирован: было подготовлено формальное описание входного языка [1, 2], была создана развернутая функциональная спецификация системы [3].

Отправной точкой проекта была публикация начальной версии нового суммировавшего накопившийся программистский опыт языка программирования, создаваемого международной рабочей группой — так называемого Алгола 58. Группа, руководимая А. П. Ершовым, стала готовить на основе Алгола 58 новый проект языка — параллельно с работавшей международной группой. Во многом направления развития языка оказались совпадающими, но в новосибирском проекте появился ряд существенно новых механизмов, поэтому в конце концов язык был сформулирован как правильное расширение окончательной версии международного языка — Алгола 60. В Альфа-языке впервые были разработаны средства, характерные для последующих за Алголом 60 языков. Было определено столь важное для вычислительных алгоритмов понятие многомерных значений, определены операции над ними, в том числе их конструирование. Были введены свойственные современным языкам концепции, такие как разнообразие видов циклов, задание начальных значений и т.п. В формальном определении языка впервые была сделана попытка выйти за пределы контекстно-свободных грамматик.

Сама работа над транслятором была в большой мере и исследовательской, и пионерской. Именно в этой работе начали складываться принципы современной оптимизирующей трансляции. Дело в том, что сам Алгол 60 был определенным вызовом для существовавших методов трансляции. Его существенно рекурсивная структура, мощность механизма процедур, возможная их вложенность и потенциальная рекурсивность, общность циклов и индексных выражений — все это требовало заметной модификации и развития техники трансляции, а самое главное — ставило вопрос о возможности получения эффективного кода. По последнему поводу высказывались сомнения, и Альфа-транслятор стал действительно первым конструктивным доказательством того, что для языков, сопоставимых по мощности с Алголом, можно построить транслятор, дающий сравнимый с ручным программированием код.

Достижению поставленной цели послужил богатый набор оптимизаций, реализованный в Альфа-трансляторе [4, 5]. Была предложена смешанная стратегия программирования (то, что позднее на Западе было названо “casing”) таких конструкций, как процедуры, циклы, индексные выражения, когда на основании анализа контекста выбирался наиболее эффективный из допустимых способ генерации конструкций. Особенно изощренно программировались процедуры и подстановка параметров

для них: выбор осуществлялся среди 11 способов. В результате алго-ловские процедуры при всей мощности механизма программировались оптимальным образом (что стимулировало программистов активно использовать это средство). Существовавшая ранее оптимизация экономии выражений была существенно развита — полностью учитывались свойства коммутативности и ассоциативности; если это преобразование не стало глобальным, то оно уже было квазилокальным. Впервые была реализована глобальная чистка циклов. Тщательной оптимизации подвергались операции над многомерными значениями — то, чего не было в стандартном Алголе, но активно использовалось вычислителями. Впервые была реализована глобальная экономия памяти, опирающаяся на теоретические работы А. П. Ершова и С. С. Лаврова.

Альфа-транслятор стал первым в мире транслятором с Алгола с большими оптимизирующими возможностями. Похожий английский проект Хоукинса и Хакстебла, который разрабатывался в это время, так и не был до конца завершен из-за сложности задачи. Конечно, успеху способствовало не только механическое соединение многих оптимизаций, но и существенное развитие существовавшей методологии оптимизации программ. Была выдвинута идея внутреннего языка, к программе на котором и применяются оптимизации, которая, хотя и не совсем в чистом виде, была реализована. Был также практически реализован глобальный анализ контекста, хотя и разрозненно для различных оптимизаций, учет которого существенно увеличил мощность оптимизаций — зародыш современного потокового анализа. Стремление практически всегда получать эффективный код привело к отказу от некоторых средств Алгола, важнейшим из которых была рекурсивность процедур.

Альфа-транслятор активно использовался во многих организациях страны. Хотя его интерфейсы с пользователем и простота эксплуатации оставляли желать много лучшего, но высокая эффективность получаемого кода обеспечила большой интерес пользователей к этой системе. Поэтому когда на смену М-20 пришла существенно более высокопроизводительная машина БЭСМ-6, организации, связанные с большими задачами, стимулировали создание версии Альфа-транслятора для БЭСМ-6. Самым быстрым решением была простая переделка Альфа-транслятора, при которой та его часть, которая отвечала за генерацию кода, была заменена генератором кода для БЭСМ-6. Возник транслятор Алгибр (“Альфа гибридный”), который был одним из первых отече-

ственных кросс-трансляторов. Такое решение имело ряд недостатков, но зато оно было реализовано быстро, и транслятор Алгibr стал первым транслятором для новой машины БЭСМ-6. Решение существенно облегчалось тем, что в Альфе-трансляторе уже существовал внутренний язык, хотя и не до конца единый для всех оптимизирующих преобразований.

Основным затруднением при Алгibr-трансляции была передача оттранслированного кода с М-20 на БЭСМ-6. В ВЦ СО АН СССР был даже разработан специальный канал передачи информации между инструментальной и объектной ЭВМ, однако это не могло быть серийным решением при массовой трансляции. Возникла необходимость “родного” транслятора для БЭСМ-6.

У разработчиков Альфа-транслятора возникала и внутренняя потребность модифицировать и улучшить как схему трансляции, так и пользовательские свойства транслятора, в первую очередь, его интерфейс с пользователем. Альфа-транслятор, несмотря на его достаточно широкую используемость, носил черты эксперимента и поиска — хотя идейные решения были найдены верно, но реализационные решения были характерны скорее для экспериментального, чем производственного транслятора (большое число просмотров программы — 24, некоторое дублирование анализа контекста, не всегда оправданное усложнение оптимизирующих преобразований). Однако не нужно забывать, что этот транслятор был первым оптимизирующим транслятором с Алгола. Кроме того, дополнительный опыт — транслятор Алгibr, незавершенный проект Альфа-транслятора для ЭВМ Урал-14 (проект ТАУ) — давали основу для новых решений.

Новый транслятор — транслятор Альфа-6 [6] — был улучшенной версией оптимизирующего транслятора с Алгола. Ряд предыдущих решений был приведен к более чистому и эффективному виду. Более четко был выделен внутренний язык и этап потокового анализа программ на внутреннем языке. Общая схема трансляции уже могла рассматриваться как типовая схема оптимизирующей трансляции (для одноязыкового транслятора), она насчитывала всего 10 просмотров. Лучшее решение была решена проблема взаимного влияния различных оптимизирующих преобразований. Идентификация и визуализация ошибок пользователя была более совершенной, что облегчало эксплуатацию. В результате система Альфа-6 стала достаточно широко используемой системой у пользователей БЭСМ-6.

АИСТ-0

В середине 60-х годов начало складываться новое направление системного программирования — операционные системы. Причины их возникновения связаны и с усложнением архитектур ЭВМ, и с совершенствованием методов взаимодействия “человек—машина”. В архитектурах появились параллельно работающие устройства, и возникла задача управления их совместной работой. Назрела необходимость автоматической смены задач, диалоговой связи с оператором ЭВМ и, что еще важнее, с пользователем. Появилось понятие режима разделения времени как средство обеспечения многозадачности. Как зарубежные, так и отечественные программисты ринулись в эту новую область, закладывая основы методологии современных операционных систем.

А. П. Ершов с его прекрасным чувством нового и перспективного стал инициатором и руководителем проекта “автоматической информационной станции” АИСТ-0 [7, 8]. Этот проект сочетал в себе создание как многопроцессорной архитектуры, так и развитого программного обеспечения, использующего те возможности, которые предоставляла эта архитектура.

АИСТ-0 стал одной из первых отечественных многопроцессорных систем. В системе существовал управляющий процессор и несколько рабочих. В качестве процессоров были взяты серийно производимые отечественные ЭВМ — управляющим процессором был взят Минск-2, а рабочим процессором — М-220. Внешняя и оперативная памяти рабочих процессоров были обобществлены и подключались через коммутатор. Для фиксации событий и для программируемых прерываний существовала система прерываний. Архитектурными и программными средствами была сохранена полная совместимость с существующим программным обеспечением для М-220.

Программное обеспечение АИСТа-0 было богатым и разнообразным, его создатели хотели использовать все предоставившиеся им возможности. Существовал ряд специализированных операционных систем пакетного режима, которые могли образовывать несколько потоков задач, исполняемых по различным дисциплинам обслуживания, учитывающих важность, срочность и прочие пользовательские требования. Были созданы многопользовательские системы диалоговой отладки и редактирования, системы диалогового программирования как для стандартных языков (Алгол 60), так и для специальных диалоговых языков (известный в то время ДЖОСС). Как пример применения диалога

в прикладных программах были разработаны система аналитических преобразований и интеллектуальный пакет для линейной алгебры. Было создано несколько информационно-поисковых систем. По-видимому, впервые в стране было разработано несколько диалоговых игровых программ.

Создание столь разнообразных систем облегчалось разработанной структурой программного обеспечения [9]. Как и другими разработчиками операционных систем разделения времени и реального времени решалась задача выделения ядра операционной системы. Такое ядро (в АИСТ-0 оно называлось диспетчер) обеспечивало создание и управление процессами, передачу сообщений между процессами, первичную обработку сигналов и прерываний, реализовывало некоторую универсальную дисциплину управления многопроцессностью, на базе которой за счет задания параметров можно было реализовать удобную для данного вида работы дисциплину обслуживания. Следующий слой программного обеспечения составляли так называемые системные программы, специализированные на способ обслуживания операционные системы. Диспетчер работал на управляющей машине, все остальное программное обеспечение — на рабочих процессорах. Такая структура облегчала обеспечение множества режимов и видов обслуживания, каждый из которых реализовался своей системной программой с использованием базовых понятий и средств диспетчера. Конечно, можно было попытаться найти общность среди некоторых классов режимов и видов обслуживания, но отсутствие реального опыта разнообразной работы делало эти попытки умозрительными, и от них сознательно отказались, сосредоточив все общие модели в диспетчере, что, по-видимому, удалось сделать довольно удачно.

Достаточно успешная реализация АИСТа-0 вызвала к жизни проект серийного производства многопроцессорных комплексов, который прорабатывался Казанским заводом ЭВМ, однако общий поворот в идеологии отечественного производства ЭВМ не дал этому проекту осуществиться. Часть идей АИСТа-0, но далеко не полностью, была использована при создании ЭВМ М-222.

Заглядывая в более поздние времена, надо отметить, что опыт, накопленный при архитектурной и программной разработке АИСТа-0, сказался и в последующих новосибирских разработках — макете потокового суперкомпьютера МАРС, первой отечественной 32-битной рабочей станции Кронос, специализированной рабочей станции для редакционно-

издательской деятельности МРАМОР. Следует отметить и дух создания новосибирских архитектур — они проводились под сильным влиянием системных программистов. Так, архитектура MAPC разрабатывалась под влиянием языка асинхронного потокового программирования Барс, а в архитектуре Кроноса были заложены средства схемной реализации конструкций языка программирования высокого уровня, прежде всего — Модулы-2.

В качестве непосредственного продолжения проекта АИСТ-0 было начато создание системы коллективного пользования для новой тогда ЭВМ БЭСМ-6, руководителем разработки программного обеспечения в котором был Г.И.Кожухин. Проект этот пришлось прекратить ввиду организационных трудностей, связанных с требуемой проектом модификации БЭСМ-6, однако он повлиял на следующий реализованный проект ВЦКП (Вычислительного Центра Коллективного Пользования), прообраза гетерогенных локальных вычислительных сетей.

Работы по АИСТ-0 не исключали и продолжение исследований и в уже традиционном для новосибирской школы направлении языков программирования и трансляции. Параллельно с работами по АИСТ-0 новосибирцами были созданы одни из первых языков системного программирования — Эпсилон и Сигма. Естественным было взаимодействие между этими работами и проектом АИСТ-0 — так, все системное программное обеспечение АИСТа-0 было написано на Эпсилоне. Языкотворчество осталось хорошей традицией новосибирской школы — позднее новосибирцами был реализован ряд интересных и оригинальных языковых проектов: язык системного программирования второго поколения Ярмо, язык программирования баз данных Бояз, язык параллельного потокового программирования Барс, язык описания и проектирования архитектуры Поляр.

Проект Бета

Накопленный в оптимизирующей трансляции опыт дал возможность перейти к следующему новосибирскому трансляторному проекту — многоязыковой транслирующей системе Бета. Цели проекта были весьма амбициозные, недаром название проекта расшифровывалось некоторыми коллегами по профессии как Большая Ершовская Трансляторная Авантюра. Предполагалось, что будет создана отрытая транслирующая система с высоким уровнем глобальной оптимизации программ, охва-

тывающая практически весь тогдашний класс императивных языков высокого уровня — от ФОРТРАНа до Алгола 68 и ориентированная на получение программ для большинства существующих архитектур. Речь шла о создании мощной базовой системы трансляции, одноязыковые трансляторы в которой выглядели как частный случай, подсистема. Забегая вперед, нужно сказать, что в принципиальном отношении задача была решена.

Для создания подобной системы определяющим было решение трех проблем — разработка общего внутреннего языка как семантического базиса широкого класса входных языков, выработка универсальной схемы языково-независимой оптимизации программ, разработка технологии включения новых входных языков в единую транслирующую систему. Все эти проблемы носили весьма серьезный характер, поэтому проект планировался как комплекс длительных методологических и экспериментальных исследований. Он выполнялся более 10 лет, задание на систему было опубликовано в 1971 г. [10], а итоговая публикация появилась только в 1982 г. [11].

Начальный этап проекта в значительной мере определялся исследованиями по внутреннему языку. Он мыслился как семантический базис, унифицирующий языковые средства входных языков, как основа оптимизации программ и как общее исходное представление программ для генерации кода. В начальных проектах внутреннего языка стремились найти ортогональные средства выражения языковых конструкций, разложив их на небольшое число семантически элементарных средств, выразить внутренний параллелизм, etc. Эти проекты были интересны в теоретическом отношении, но, к сожалению, оказались непрактичны. Окончательная версия внутреннего языка была создана как объединение абстракций общих понятий и конструкций широкого спектра входных языков с включением не полностью интерпретируемых языково-зависимых конструкций. Версия эта была построена на анализе большого числа существовавших тогда языков программирования — Алгола 60, ПЛ/1, Симулы-67, Алгола 68, Паскаля (ФОРТРАН игнорировался из-за наличия в нем рудиментарных конструкций, Кобол же по существу никогда не был в поле зрения отечественных программистов), на выделении общностей этих языков — в системе типов, в управляющих конструкциях, на построении абстракций, как отражающих эти общности, так и дающих простор для включения специфических средств данного языка. Вся эта работа была частью общемировых исследований по внут-

ренным языкам и помимо своей конструктивной роли для Бета-проекта обогатила понимание общей содержательной семантики существующих языков программирования.

Оптимизирующая трансляция в системе БЕТА реализуется специальным оптимизирующим процессором, осуществляющим глобальные оптимизирующие преобразования программы на внутреннем языке. Оптимизирующее преобразование понимается как пара — условие применимости преобразования и собственно схема трансформации в терминах внутреннего языка. Условия применимости большинства преобразований существенно используют управляющие и информационные зависимости между операторами программы, поэтому в оптимизирующем процессоре выделяется начальный этап потокового анализа. В результате его работы над операторами внутреннего языка надстраивается гамачно-цикловая иерархическая структура, а каждый оператор и компонента такой структуры снабжаются списками аргументов и результатов. Как условия применимости, так и трансформации определяются в терминах введенной иерархии, кроме того, иерархия используется для факторизации глобальных оптимизаций.

Собственно трансформирующая часть оптимизирующего процессора построена как открытый набор преобразований. Был выбран некоторый порядок их исполнения, дающий наибольший эффект их совместного применения с учетом их взаимозависимости. Ряд оптимизаций был существенно обобщен — чистка циклов реализовывалась не только за счет выноса перед циклом фрагментов выражений, но и за счет выноса как перед, так и после цикла операторов целиком, а также целых гамаков; максимально осуществлялось удаление несущественных операторов, обобщавшее так называемую ликвидацию “мертвого кода”.

Таким образом, в системе БЕТА были развиты в сторону кристаллизации те понятия оптимизирующей трансляции, которые были найдены в Альфа-проектах: внутренний язык стал действительно единым, потоковый анализ был выделен как отдельный фрагмент, да и само выполнение языково-независимых оптимизаций было выделено в отдельную компоненту транслирующей системы, которая могла привлекаться по желанию пользователей.

Технологической целью проекта была максимальная унификация трансляции с различных языков. Общая схема трансляции включала двухпроходную схему получения абстрактной программы на внутреннем языке (декомпозиции), охватывающую лексический, синтаксиче-

ский и контекстный анализ (язык Ада внес сюда некоторые коррективы), многопроходную работу оптимизирующего процессора, включающую лишь по специальному запросу, и двухпроходную схему генерации кода. Оптимизирующий процессор не зависит ни от входного, ни от выходного языка — языково-зависимые конструкции внутреннего языка снабжались нужными для оптимизации атрибутами и были в этом смысле частично интерпретируемы. Вначале предполагалось, что как получение абстрактной программы, так и генерация кода будут строиться специальными метапроцессорами по описанию входного и выходного языков соответственно, однако в окончательной версии победил модульный подход. Была сделана библиотека модулей декомпозиции, фактически вводящая обобщенные операции, в терминах которых строились лексический, синтаксический и контекстный анализ для конкретного языка. Точно также была создана библиотека генерации кода, включавшая модули общих для генерации действий — обхода программы на внутреннем языке, назначения регистров (с учетом конкретных параметров архитектуры) и пр. Как показал опыт, общая часть декомпозиции и генерации кода (оптимизация была универсальной) составляла в зависимости от языка от 40% (Ада) до 60% (Паскаль).

В целом этот проект был интересным экспериментом в трансляции программ, по своему размаху превосходящим другие опыты многоязыковой трансляции, существовавшие в мире. Были реализованы Симула 67, Паскаль, Ада и Модуля-2, причем два последних языка, не участвовавшие в выработке схемы трансляции и внутреннего языка, достаточно хорошо вписались в систему, что свидетельствовало о надежности принятых в проекте решений. Выходными языками были столь разные языки БЭСМ-6 и СМ-4.

Область трансляции стала традиционной для новосибирцев. Было реализовано множество трансляторов для различных языков — от Сетла до Оберона, традиции многоязыковой трансляции были продолжены в биязыковой системе программирования XDS.

Теория

Наряду с исследованиями в различных направлениях системного программирования — и это очень важно отметить — в новосибирской школе велись исследования и по теоретическому программированию, основы которых были заложены А. П. Ершовым. Сознательная его позиция, заключавшаяся в том, что серьезные эксперименты по созданию

систем должны подкрепляться разработкой теоретических моделей и стимулировать эту разработку, была весьма плодотворной.

Опыт в трансляторных проектах вызвал естественный интерес к теории схем программ — как последовательных, так и параллельных. Первыми работами новосибирской школы в этом направлении были работы А. П. Ершова по операторным алгоритмам, прообразам стандартных схем, и по аксиоматике схем Янова, когда предложенная Яновым аксиоматика преобразований была сведена к существенно меньшему числу аксиом. Работы по схемам программ концентрировались вокруг проблем эквивалентности и преобразований схем Янова. А. П. Ершовым и его учениками (В. Э. Иткиным, В. К. Сабельфельдом, Э. Л. Горель) был получен ряд интересных результатов [12]: был найден общий критерий локальности правил преобразований схем Янова, доказана логическая независимость правил преобразования, построена полная система преобразований с отношением тождества, доказана распознаваемость логико-термальной эквивалентности и пр.

Новосибирская школа одной из первых в мире занялась проблемами теории параллельного программирования. В. Е. Котовым и А. С. Нариньяни была предложена одна из первых моделей параллельных программ — недетерминированные асинхронные автоматы. Для этой модели получен ряд важных результатов, связанных с условиями отсутствия конфликтов параллелизма, с преобразованием последовательных программ в параллельные и преобразованиями, увеличивающими внутренний параллелизм — это определяющие проблемы теории параллелизма.

Для коллектива, активно работающего в области трансляции, естественным было обратить внимание на теорию оптимизации программ и на применение этой теории для создания оптимизирующих трансляторов. А. П. Ершовым (совместно с С. С. Лавровым) были заложены здесь основы современной теории экономии памяти [13]. Результаты, полученные при этом, были использованы при построении оптимизирующей части Альфа-транслятора. Был предложен ряд моделей программ, ориентированных на обоснование других оптимизирующих преобразований [14]. Модель линейных схем позволяла обосновывать преобразования, связанные с удалением и перестановкой операторов, что было использовано в системе БЕТА. Модель линейных программ позволила построить оптимальный алгоритм преобразования линейных участков. Была предложена и более общая модель крупноблочных схем, объединяющая преимущества линейных схем и преобразований, связанных с

памятью.

Так закладывалось в новосибирской школе направление исследований по теоретическому программированию. Впоследствии эти работы вылились в исследования по специализации программ — А. П. Ершовым был предложен принцип смешанных вычислений, который дал толчок потоку исследований разных коллективов мира, была предложена идея конкретизации программ — по дальнейшим исследованиям в теории оптимизации программ и в моделях параллельных программ, по спецификации и верификации параллельных и распределенных программ.

Искусственный интеллект

А. П. Ершов стимулировал возникновение и еще одного направления работ новосибирской школы — искусственного интеллекта. Г. И. Кожухиным еще в 1959 г. была построена, по-видимому, одна из первых программ по искусственному интеллекту — она на основании знаний путем проб и ошибок строила алгоритм нахождения корней полинома. Дальнейшие исследования по искусственному интеллекту возглавлялись А. С. Нариньяни, учеником А. П. Ершова. Вначале эти исследования концентрировались вокруг проблем понимания естественного языка и недетерминированного управления шагающим роботом. Решение задач, возникавших здесь, привело к созданию современных направлений исследований — недоопределенности как основе программирования в ограничениях и методологии создания инструментов построения интеллектуальных систем.

Эта триада — системное программирование, теоретическое программирование и искусственный интеллект — с их взаимным влиянием и остается полем деятельности новосибирской школы. Основы всей этой деятельности возникли при могучем влиянии такого выдающегося отечественного ученого, как академик А. П. Ершов.

Цитируемые источники

1. **Ершов А. П., Кожухин Г. И., Волошин Ю. М.** Входной язык системы автоматического программирования: предварительное сообщение. — М.: ВЦ АН СССР, 1961. — 176 с.
2. **Ershov A. P., Kozhuhin G. I., Voloshin Yu. M.** Input language for automatic programming systems. — Acad. Press, London—New York, 1963. — 70 p.

3. **Ершов А. П.** Основные принципы построения программирующей программы Института математики Сибирского отделения Академии наук СССР // Сиб.мат. журнал. — 1961. — Т. 2, № 6. — С. 835–852.
4. **АЛЬФА-система** автоматизации программирования / Под. ред. А. П. Ершова). — РАН. Сиб.отд-ние. — Новосибирск: Наука, 1967. — 308 с.
5. **The Alpha automatic programming system** / (Ed. by A.P.Ershov). — Acad. Press, London—New York, 1971. — 247 p.
6. **Аникеева И. Н., Богданова С. Ф., Буда А. О. и др.** Система автоматизации программирования АЛЬФА-6 // Системное программирование. — Ч. 1. — Новосибирск, ВЦ СО АН СССР, 1973. — С. 12–23.
7. **Ershov A. P., Kozhuhin G. I., Makarov G. P., Nечepurenko M. I., Pottosin I. V.** An experimental automatic information station AIST-0 // Proc. of AFIPS Conf.— Vol. 30.— Academic Press, Washington-London, 1966.— P. 577–582.
8. **Вишневский Ю. Л., Ершов А. П., Кожухин Г. И. и др.** Экспериментальная система коллективного пользования АИСТ-0 // Тр. 2-ой Всесоюзной конф. по программированию. Заседание Н. — Новосибирск, ВЦ СО АН СССР, 1970. — С. 3–14.
9. **Поттосин И. В.** Структура операционных систем коллективного пользования // Некоторые проблемы вычислительной и прикладной математики. — Новосибирск: Наука, 1975.
10. **Ершов А. П.** Универсальный программирующий процессор // Проблемы прикладной математики и механики. — М.: Наука, 1971.— С. 103–116.
11. **Ершов А. П., Касьянов В. Н., Покровский С. Б., Поттосин И. В., Степанов Г. Г.** Методика разработки многоязыковых трансляторов на примере системы БЕТА // Математическая теория и практика систем программного обеспечения. — Новосибирск, ВЦ СО АН СССР, 1982. — С. 64–80.
12. **Ершов А. П.** Современное состояние теории схем программ // Проблемы кибернетики. — Вып. 27. — М.: Наука, 1973. — С. 87–110.
13. **Ershov A. P.** Axiomatics for memory allocation // Acta Informatica. — 1976. — Vol. 6, № 1. — P. 61–75.
14. **Поттосин И. В.** О математических моделях программ, ориентированных на оптимизацию программ // Proc. VII Nation. School “Mathematical Methods in Informatics”. — Varna, 1981.

Е. А. Жоголев

КАК А. П. ЕРШОВ СТАЛ ПРОГРАММИСТОМ

А. П. Ершов окончил механико-математический факультет МГУ по кафедре Вычислительной математики в 1954 году. Это был первый выпуск кафедры, для которого читался курс программирования. Лекции читал профессор А. А. Ляпунов.

Кафедра Вычислительной математики была создана на мехмате МГУ в 1949 году. В 1950 году на ней защитили дипломные работы два или три выпускника, а в 1951 году был выпуск этой кафедры по сокращенному учебному плану. Я был среди выпускников этой кафедры 1952 года, которые закончили ее по полному учебному плану. Но курса программирования нам еще не читали. Был у нас только курс по электронным схемам вычислительных машин.

В 1951 году, когда А. П. Ершов заканчивал 2-й курс, он должен был выбрать кафедру для продолжения обучения. И А. П. Ершов записался на кафедру Высшей алгебры. Но в это дело вмешался счастливый случай.

С 1949 года в МГУ регулярно проводилась традиционная легкоатлетическая эстафета на приз газеты “Московский университет”. Дистанция эстафеты проходила от Киевского вокзала до строительства нового здания университета на Ленинских горах и состояла из 14 этапов. Сначала эта эстафета проводилась в последнее воскресенье апреля (в частности, и в 1951 году, хотя в работе [1] ошибочно указана дата 5 мая), позже она стала проводиться в первое воскресенье мая. В 1951 году Андрей и я бежали за первую команду мехмата. После эстафеты мы вместе с ним пошли пешком к Киевскому вокзалу. Он стал расспрашивать меня о нашей кафедре. Зав. кафедрой Вычислительной математики в то время был профессор Б. М. Щиголев (в работе [1], опять-таки ошибочно, указан академик С. Л. Соболев — он стал зав. кафедрой только после переезда в новое здание). Я рассказал Андрею все, что знал об электронных вычислительных машинах (я в то время был уже на 4-м курсе). Особенно большое впечатление на Андрея произвела память на ртутных трубках. Потом мы долго фантазировали о возможностях этих машин. И так были увлечены разговором, что прошли мимо Киевско-

го вокзала и дошли пешком аж до самого общежития университета на Стромынке (на берегу реки Яуза).

После этого разговора А. П. Ершов забрал свое заявление с кафедры Высшей алгебры и подал его на нашу кафедру. Много лет спустя, в 1974 году, даря мне сборник переводов под его редакцией [2], он назовет меня своим “крестным отцом”.

И еще я вспоминаю о его студенческой свадьбе в общежитии, для которой я написал четверостишие (см. ниже). Оно до Андрея так и не дошло (я редко отдавал свои стихи адресатам).

Свадебное слово

На свадьбу А. П. Ершова

Священный сей союз благословлен —
Слились их души воедино.
Два сердца бьются в унисон:
Андрея сердце, сердце Нины.

Октябрь 1952

После окончания университета мы с А. П. Ершовым регулярно встречались на различных программистских форумах. Он стал признанным лидером программистов СССР. Но этот период его деятельности хорошо известен, многие об этом уже высказывались. И я вряд ли могу сказать лучше. Но хочу лишь отметить его неизменно доброжелательное отношение к моим выступлениям, далеко не всегда удачным.

Список литературы

1. Карл Левитин. Прощание с АЛГОЛом. — М.: Знание, 1989.
2. Как проектируются и создаются программные комплексы. Сб. переводов под ред. чл.-корр. АН СССР А.П. Ершова. — М.: ИТМ и ВТ АН СССР, 1974.

Л. Л. Змиевская
ЧУТЬ ВОСПОМИНАНИЙ...

В моей трудовой жизни были работа и ситуация, о которых я никогда не смогу забыть.

Все началось осенью 1959 года. Наш отдел программирования, возглавляемый Андреем Петровичем Ершовым, территориально размещался в Институте математики им. Стеклова АН СССР. Там нам было выделено несколько комнат, рядом с кабинетом С. Л. Соболева. В этом же здании на 1-ом этаже располагался Вычислительный центр, обслуживавший наряду с сотрудниками Стекловки и нас.

В отдел обратился заказчик, сейчас уж не помню кто, с просьбой найти обратную матрицу для заданной матрицы 436 порядка.

Выбор пал на меня и Вадима Янкова. Исходная матрица была “дырявой”, как сито: очень много нулевых элементов. В поиске алгоритма мы с Вадимом остановились на алгоритме обращения матриц, предложенном в свое время Андреем Петровичем.

Первый этап — написание программы и ее отладка — занял не слишком много времени. На этом этапе работу вели я и Вадим Янков.

Второй этап — подготовка данных — осуществляла уже я одна. Это была трудоемкая работа, долгая и нудная. Тогда я научилась и “читать”, и править информацию на перфоленте: на месте с ошибкой недостающую дырочку пробивать, лишнюю — заклеивать. Главное — точно локализовать то место, куда следовало приложить руку в прямом смысле этого слова.

Третий этап — собственно счет по отлаженной программе — начался где-то весной 1960 года.

В январе того же года я вышла замуж, и когда пришел месяц июль, я решила поехать в первый раз к родителям мужа. Им не терпелось со мной познакомиться.

Андрей Петрович в это время был в отъезде. С разрешения того, кто его заменял по Отделу (кажется, Гены Кожухина), я с чистой совестью уехала отдыхать. К тому времени было сделано несколько десятков “шагов” по пути приближения к желанной обратной матрице.

В положенное время я появилась на работе. И тут грянул гром! Андрей Петрович узнав, что матрица все еще не обращена, что вместо того, чтобы в поте лица трудиться, я разъезжаю по гостям, он в первый (и слава Богу в последний!) раз в моей жизни, не повышая голоса (но металл в голосе был явно!), сделал строгий выговор за то, что личные интересы я поставила выше производственных! И это несмотря на симпатию, которую я всегда чувствовала по отношению к себе.

Слезы градом покатались из моих глаз! Извинившись, я убежала, умылась, успокоилась и, вернувшись, пообещала приложить все силы, чтобы как можно скорее закончить обращение злополучной матрицы.

Счет шел на отечественной машине БЭСМ-2. Хотя она для того времени и была на высоте, но ее быстродействия и надежности для решения больших задач было, конечно же, недостаточно.

Времени просили много. Его давали, но как правило ночью. Благо, жили мы тогда уже в Черемушках, до работы пешком минут 15–20. Время в стране было спокойное. Так что и прийти на работу, и уйти домой ночью было не проблема.

Любо-дорого вспоминать: сидишь за пультом ночью одна в огромном зале, магнитные ленты крутятся, лампочки тебе подмигивают. В комнате рядом с машинным залом дежурные инженеры. Они развлекают себя игрой или просто спят. Когда машина сбивается, не паникуешь, а пускаешь тесты и призываешь на помощь инженеров, часто показывая ту ячейку, которую надо заменить.

“Быстро сказка сказывается, да не скоро дело делается!”

В октябре я поняла, что в июне 1961 года стану мамой. Все наши уезжали в Академгородок с намерением заняться транслятором с Алгола-60, изучение которого началось еще в Москве на семинарах Отдела. А я и Софья Давыдовна Тартаковская, определенная мне в помощники, оставались в Москве — заканчивать обращение надоевшей матрицы. Ничего нового и интересного в этой работе уже не осталось, надо было только добить, дожать до конца все 436 итераций.

В начале апреля 1961 года я ушла в декретный отпуск. К тому времени из 436 итераций были просчитаны более 410.

Но “чем дальше в лес, тем больше дров”! Каждая следующая итерация давалась с большим трудом — машина сбивалась, и опять надо было повторять просчет. Не была готова БЭСМ-2 ни по памяти, ни по быстродействию для решения таких громоздких задач. Важно было иметь длительный период стабильной работы, без сбоев.

Закрывать амбразуру собой пришлось одной Софье Давыдовне. В конце 1961 года, успешно завершив давно и не ею начатую работу, она наконец-то смогла уехать в Академгородок.

А я появилась в Академгородке с маленьким сынишкой весной 1962 года. Работа над Альфа-транслятором шла полным ходом. Но и для меня нашлось дело, чему я была несказанно рада.

Конечно же, Андрей Петрович не допускал и мысли, что работа по обращению той матрицы 436 порядка окажется такой продолжительной! Он думал, что еще пару месяцев, июль–август 1960 года, и результат будет достигнут. С такими мыслями он имел полное моральное право на тот тон и те слова!

Я на Андрея Петровича за давние горячие слезы не в обиде! Они помогли мне осознать, что, работая в коллективе, всегда надо помнить о чувстве долга и ответственности перед ним.

И. В. Поттосин

ИСТОРИЯ АЛЬФА-ПРОЕКТА

Роль Альфа-проекта

Альфа-проект занимает особое место в истории новосибирской школы программирования. Именно с него начинается всесоюзная и мировая известность этой школы. Это была первая работа, которой мы заявили о себе как о сильном коллективе, умеющем решать сложные и нужные задачи. Это был первый в мире оптимизирующий транслятор с языков типа Алгол.

Конечно, и до этого существовали оптимизирующие трансляторы: и отечественные ПП (программирующие программы), и зарубежные Фортран-трансляторы, но они оптимизировали программы, написанные на куда более простых и приспособленных для такой оптимизации языках. Алгол 60 разрабатывался как “естественный математический” язык, без внимания к получению эффективных программ. Мало того, некоторые молчаливо предполагали, что за естественность формулирования алгоритмов следует платить недостаточной эквивалентностью. Это давало возможность некоторым отечественным программистам говорить на программистских форумах, что Алгол 60 — это западная провокация, предназначенная для того, чтобы завлечь нас в тупик: “Они-то будут писать на Фортране или мнемосодах, а мы, поддавшись влиянию Запада, будем создавать неэффективные программы”. Так действительно было.

Алгол 60 создавал ряд новых, неизведанных проблем для трансляции, и решению этих, собственно трансляционных, проблем и было посвящено большинство западных и отечественных реализаций Алгола 60. Многие считали, что создание Алгол-трансляторов, дающих достаточно эффективный код, невозможно, и в процессе нашей работы мы неоднократно и на разных уровнях слышали, что взялись за неразрешимую задачу. Надо честно сказать, что сложность задачи мы недооценили, сроки, которые мы намечали вначале, не выдерживались, и наше непосредственное начальство, считая проект нереализуемым, пыталось его прекратить. Только личная поддержка С. Л. Соболева, нашего директора, спасла нас. Ситуация усугублялась тем, что мы реализовывали

не Алгол 60, а его расширение, которое включало конструкции, существенно усложнявшие как оптимизацию, так и собственно трансляцию.

Успешная, хоть и затянувшаяся, реализация проекта была конструктивным доказательством того, что с языков типа Алгол возможна эффективная реализация, что естественность конструкций не противоречит порождению эффективного кода, но требует изощренной техники оптимизации. Начало создания такой техники и было положено в Альфа-проекте, хотя она и опиралась на предыдущие работы по ПП и Фортран-трансляции.

Альфа-язык

В 1958 году международная группа опубликовала начальную версию нового языка программирования, основной идеей которого было дать как можно более естественную форму для выражения алгоритмов (прежде всего, вычислительной математики). Опираясь на эту версию, Андрей Петрович Ершов вместе с Геной Кожухиным и Димой Янковым решили разработать свой язык, внося в него то, что было бы удобно вычислителям, в частности, возможность оперировать с векторами, матрицами и многомерными объектами. Когда своя версия была готова, появилась окончательная версия международного языка Алгол 60. Ряд решений в Алголе 60 совпадали с принятыми нашей группой, но ряд существенных конструкций, которые были разработаны в “сибирском” языке, отсутствовали в Алголе 60. Было принято решение удалить все несущественные различия из “сибирского” языка, но существенные — оставить, сделав из него правильное расширение Алгола 60. Дима Янков к этому времени ушел из института, и работу по созданию окончательной версии языка выполнила группа в составе Ершова, Кожухина (автора первого перевода Алгола 60 на русский язык) и вновь пришедшего в институт Юлика Волошина.

Новый язык в разное время носил разные названия. Сначала он носил название “входной” (в английском переводе Input Language), затем — скорее обиходное, чем официальное — “сибирский” и, наконец, уже установившееся и официальное имя — язык Альфа.

Как уже говорилось, Альфа-язык был расширением Алгола 60, более того, расширением некоторого его подмножества. Основные ограничения этого подмножества были связаны с главной задачей — обеспечить высокоэффективный код. Все, что мешало эффективной генерации кода или требовало сложного анализа при проведении оптимиза-

ций, удалялось. Так, требовалась обязательная спецификация параметров-меток, запрещались именуемые выражения и идентификаторы стандартных функций как фактические параметры, ограничивалось использование переменных с индексом в качестве параметров циклов, при неопределенном значении указателя переключателя переход был неопределен и вводились другие тому подобные ограничения, которые отсекали всякую трудно реализуемую и трудно понимаемую экзотику. По-настоящему серьезных ограничений было два: были запрещены рекурсивные процедуры, так как мы не нашли хорошего метода реализации таких процедур, а реализовать каждую процедуру как потенциально рекурсивную (что необходимо, если не проводить сложного анализа) было бы накладно; побочный эффект вызовов функций игнорировался: мы хотели избавиться от сложного анализа, когда он влияет на корректность оптимизаций, а когда не влияет.

Наряду с этими, не так уж серьезными (кроме отказа от рекурсивности), ограничениями, Альфа-язык содержал ряд принципиальных новшеств. Большинство из них увеличивали выразительную силу языка для вычислительных задач.

Впервые для языков программирования арифметические типы были дополнены типом комплексный с введением всей комплексной арифметики. Также впервые было введено понятие многомерного значения (массива значений) и многомерной переменной, обладающей этим значением целиком — до сих пор в языках действия с массивами должны были трактоваться как некоторая совокупность действий над их компонентами, в Альфа-языке появилась крайне удобная и широко использовавшаяся возможность выполнять естественные операции над всем многомерным значением целиком и целиком же присваивать это значение многомерной переменной. Нововведение было явно выигрышным и в ряде последующих языков — Алголе 68, ПЛ/1, Аде — были введены подобные же средства. В языке появились средства создания многомерных объектов — значений и переменных: образования подмассивов таких объектов (“вырезок” в терминах последующих языков), объединения объектов в новый многомерный объект (с повышением размерности или без). Последние средства, так называемые геометрические операции, несмотря на свою полезность, так по-настоящему и не прижились в последующих языках — исключением, пожалуй, тут является язык Мучника и Шафаренко Eval. По-видимому, причина здесь в том, что для полной и естественной реализации подобных операций объеди-

нения (геометрических операций) надо глубокое проникновение семантики, связанной с многомерными значениями с некоторой структурой, в язык, что было характерно и для Альфа, и для Eval.

В вычислительных задачах зачастую используются такие механизмы, как рекуррентность и итерации. Эти механизмы характерны тем, что надо иметь несколько последовательных (упорядоченных, так сказать, по времени) экземпляров объекта и при каждом получении нового объекта сдвигать эту последовательность. Для естественного выражения итерации и рекуррентности в Альфа-язык было введено ни в каком языке более не повторенное понятие объектов с верхним (временным) индексом, для которых операции по заведению последовательности и ее сдвигу реализовывались по умолчанию.

Впервые в Альфа-языке были введены циклы без параметра и описания начальных значений, что сейчас присутствует во многих языках. Из Фортрана было позаимствовано такое наглядное средство, как функции-выражения (тот простой случай, когда значение функции определяется единственным выражением). В качестве способов подстановки параметров процедуры была добавлена подстановка результатом (out-параметр Ады). Логические выражения могли включать цепочки равенств или неравенств (вида $a < b <= c < d = e$).

Еще одно нововведение в Альфа-языке — это перечисления. В целом ряде конструкций языка — цепочке неравенств, цепочке присваиваний, цепочке описания начальных значений, геометрических операциях — допускались перечисления, причем во всей их полноте (например, $A[i+1,j] := \dots := a [i+1,m] := 0$).

Интересно, что для задания синтаксиса перечислений пришлось прибегнуть к контекстно-зависимым правилам грамматик — это был первый пример использования контекстно-зависимой грамматики для описания синтаксиса языка (Алгол 68, конечно, в большей мере использовал механизмы контекстно-зависимых грамматик, но он был позже).

Конкретный входной язык системы Альфа содержал еще ряд дополнений Альфа-языка, учитывавших, в том числе, и конкретную архитектуру объектной машины (М-20 или БЭСМ-6), и сложившийся состав программного обеспечения. Для систем Альфа и Альфа-6 эти дополнения несколько различаются, в дальнейшем я буду говорить только о входном языке системы Альфа.

Одним из важных дополнений было введение библиотеки процедур. Насколько мне известно, в системе Альфа впервые было введено такое

понятие, которое в современных системах выражается библиотеками модулей, пакетов, классов. Состав библиотек никак не стандартизировался, он мог быть определен при установке системы, для этого существовала своя технология. Те процедуры из библиотеки, вызов которых встречался в программе, подключались, как это сейчас принято, к ней как объекты самого внешнего контекста.

Надо заметить, что нам было легко пойти на это новшество: процедуры реализовывались Альфа-системой максимально эффективным образом, а влияние их вызовов на реализацию вызывающей программы учитывалось настолько адекватно, насколько возможно. Разумеется, это достигалось не только за счет изощренного программирования процедур, но и за счет тех ограничений Алгола 60 (рекурсивность и побочный эффект), на которые мы сознательно пошли. В результате пользователи системы могли всю пользоваться таким могучим изобразительным средством, как процедуры, не опасаясь, что это ухудшит их эффективность рабочих программ. В качестве контраста вспоминается, что в руководстве по пользованию другой системой программирования для Алгола 60, где процедуры реализовывались во всей своей мощности, в рекомендациях пользователям указывалось, что во избежание существенного ухудшения эффективности пользоваться процедурами не рекомендуется. Что поделаешь, техника программирования процедур тогда еще делала первые шаги, а сами процедуры в Алголе 60 несли ряд неконструктивных, плохо реализуемых черт (например, подстановка именем), от которых последующие языки избавились.

Учет архитектуры и программного обеспечения реализовывался такими средствами входного языка, как использование библиотеки ИС-2 (набор подпрограмм в машинном коде, динамически вызываемых и связываемых с рабочей программой) и машинных команд. Синтаксически обращение к стандартной подпрограмме из библиотеки носило вид вызова процедуры, а так как состав библиотеки был фиксирован, то компилятор знал свойства данной подпрограммы, нужные для потокового анализа. Позволялось использовать единичные машинные команды как операторы входного языка. Так как эти операторы включались в общий потоковый анализ и не должны были мешать оптимальной генерации кода, то их употребление должно было подчиняться некоторой дисциплине.

Так как оперативная память ЭВМ М-20 была маленькой (сейчас это выглядит совсем удивительно — всего 4К 45-разрядных слов), то надо

было ввести механизмы, поддерживающие работу программ с довольно большими требованиями на память. В системе программирования ТА-2 (тоже для М-20) решением стало создание виртуальной памяти, что не требовало никаких усилий от пользователей, но влекло накладные расходы. Мы пошли по другому пути: заставили пользователя “укладываться” в оперативную память, самостоятельно фрагментируя программу и данные с выделением внешних (располагаемых во внешней памяти — для М-20 это барабан и ленты) массивов и блоков. Внешние блоки скачивались с внешней памяти и стирались по мере необходимости автоматически (run-time системой), а внешние массивы надо было явно копировать в буферные массивы в оперативной памяти (или из них).

Последним по времени новшеством во входном языке было введение средств комплексации. Комплексовались такие программы, каждая из которых занимала всю оперативную память, а все информационные связи между ними могли осуществляться только через внешние массивы. Все программы в комплексе нумеровались, и этот фиксированный номер использовался для явного указания динамического преемника данной программы комплекса. Комплексование позволяло создавать сложные (по тем временам) программные системы, и его необходимость мы осознали уже после того, как система достаточно широко пошла по стране. Основным толчком к введению комплексования была здоровая критика вычислителей из Отдела прикладной математики (теперь ИПМ РАН им. М. В. Келдыша) — Г. Лимана, Л. Майорова, М. Степановой и Л. Фрязиновой. Они приехали из Москвы со своеобразной инспекцией — посмотреть, насколько Альфа-система подходит для их больших задач (а у них в ОПМ тогда действительно были очень большие по тем временам вычислительные задачи). Свои эксперименты с системой они проводили в очень дружественной атмосфере, качество получаемого кода им весьма понравилось, а вот в то Прокрустово ложе, связанное с ограниченной памятью, их серьезные задачи не вкладывались. Такие живые примеры нас убедили, и (чего не сделаешь для друзей) мы пошли на нужные расширения языка и развитие системы.

Разработка Альфа-транслятора

Создание Альфа-транслятора началось с того, что А. П. Ершовым был разработан общий проект транслятора и выделены проблемы, которые надо было решить. В первой половине 1961 года такой проект был готов.

Затем надо было сформировать коллектив разработчиков. Тут учитывалось и желание, и возможности. Не все из отдела хотели браться за эту неподъемную работу, не все реально могли ее выполнять. У отдела были и другие задачи, нельзя было их оголять. В результате сформировался совершенно молодежный и практически не имеющий опыта (только трое имели дело с трансляцией до этого) коллектив — это были или сравнительно недавние выпускники университетов или только что кончившие университет: Ершов, Змиевская, Трохан (Московский университет), Бабецкий, Кожухин, Потгосин (Томский университет), Кожухина, Мишкович (Ленинградский университет), Волошин (Рижский университет), Бежанова (Горьковский университет), Загацкий (Саратовский университет), Михалевич (курсы программистов СО АН СССР, потом Новосибирский университет). Компания разноплеменная и не очень искушенная в программировании, особенно для такого сложного, пионерского проекта (средний стаж в программировании в среднем был три с половиной года: от 8 лет до года, большинство — 2-3 года). Все искупало два фактора: хорошая организация работ, которую проводил Андрей Петрович, и общее желание сделать “большое дело”.

Все создание Альфа-транслятора было разбито на следующие этапы.

- Создание проектного задания. Это было сделано Ершовым при участии Кожухина.
- Продумывание алгоритмов трансляции и оптимизации. В этом участвовали практически все разработчики. За каждым был закреплен какой-либо “участок”, в котором он должен найти решение, обсуждавшееся затем сообща. В результате была выработана последовательность действий (блоков транслятора) и представление об общих структурах данных, которые связывали блоки. Блок обычно поручался тому, кто продумывал соответствующие алгоритмы.
- Разработка блоков. Алгоритм блока специфицировался на специальном неформальном, но достаточно точном языке.
- “Ручная” прокрутка блоков. Это было специальное действие. Разработчик блока и кто-либо прикрепленный (из числа других разработчиков) садились рядом и “исполняли” спецификацию на некотором примере. Результат исполнения использовался при прокрутке следующего блока. Прокрутка шла последовательно по блокам транслятора, в соответствии с их следованием.

- Программирование. По разработанной спецификации писалась машинная программа.
- Автономная отладка блоков. Она шла параллельно для всех блоков, использовались примеры прокрутки.
- Комплексная отладка. Здесь возникали проблемы, которые приводили к перепроектированию некоторых блоков, — прежде всего, при обнаружении слишком больших временных затрат.

Общая структура организации была такой — руководитель Ершов, два координатора — Кожухин (1-я часть транслятора) и Поттосин (2-я часть транслятора). Руководитель отвечал за принятие всех основных решений. Предполагалось, что он свободен от собственно разработки и принятия технических решений, но практически он участвовал в разработке блоков экономии памяти, а кроме того, когда выяснилось, что нужен специальный блок превращения промежуточного представления программы во внутренний язык, а все разработчики уже “разобраны” по блокам, то Ершову пришлось самому писать этот блок. Координаторы отвечали за проблемы, возникавшие у разработчиков блоков и следили за согласованием работ. Особую роль играл Волошин. Он зачастую в ущерб своей работе (в результате его блоки задержались) участвовал в обсуждении многих возникавших вопросов и часто помогал неожиданным взглядом на проблему. Польза такого “блуждающего дискуссионта” в проекте несомненна.

Состав Альфа-транслятора и персонала

Общая структура транслятора состоит из четырех этапов: получение промежуточного представления, превращение его во внутренний язык и оптимизации на внутреннем языке, построение машинных команд, экономия памяти и формирование машинной программы. Все эти этапы разбивались на блоки. Разбиение определялось как функциональными действиями, так и необходимостью “вписаться” в малую (4К слов) память компьютера.

Окончательная структура блоков такова:

- Ввод и первичный синтаксический контроль (блок 1), разработчик и реализатор — Загацкий.
- Обработка описаний (блок 2) — Кожухин.
- Завершение идентификации (блок 3) — Бабецкий.
- Декомпозиция выражений (блоки 4 и 5), разработчик — Кожухин, реализатор — Кожухина.

- Анализ и программирование процедур (блоки 6 и 7) – Загацкий.
- Перевод из промежуточного представления во внутренний язык (блок 8) — Ершов.
- Обработка геометрических операций над массивами и реализация действий с многомерными массивами (блоки 9, 10) — Волошин.
- Анализ и чистка циклов (блоки 11-13) — Бежанова (при участии в разработке Поттосина).
- Экономия выражений (блок 14) — Поттосин.
- Построение машинных команд (блок 15) — Кожухина (при участии в разработке Бабецкого).
- Программирование индекс-регистров и циклов (блоки 16-18) — разработчик Поттосин, реализатор Кожухина.
- Экономия констант (блок 19) — Змиевская (при участии в разработке Ершова).
- Построение операторной схемы и графа несовместимости (блоки 20, 21) — Мишкович (при участии в разработке Ершова).
- Распределение памяти (блок 22) — Трохан (при участии в разработке Ершова).
- Чистка и компоновка программы (блоки 23, 24) — Трохан, Змиевская.

Доводка и внедрение

Когда Альфа-транслятор начал “дышать”, начали проявляться и не предвиденные ранее проблемы. Насколько помнится, особенно болезненными были три.

Вначале мы не предполагали никакой нейтрализации ошибок: транслятор останавливался на первой найденной (зачем транслировать далее неправильную программу?). Наши пользователи тут же застонали: надо было столько раз повторять трансляцию, сколько ошибок в программе. После этого была введена нейтрализация — пропуск до первого опорного символа (начало, конец, точка с запятой).

Обнаружилось, что при большом числе различных идентификаторов наш лексический анализ “жрет” большую часть времени трансляции, пришлось вводить функцию расстановки для отождествления имен.

Выяснилось, что при возрастании размера программы время на экономии памяти резко возрастает и на достаточно больших программах занимает львиную долю времени трансляции. Пришлось пересмотреть

алгоритм и переконструировать структуры данных.

Доводка транслятора шла в “боевом” режиме, на реальных задачах Н. Ершовой, В. Каткова, С. Ривина, Е. Каленковича. Особенно плодотворно (для нас, разработчиков) шла работа с задачами Владика Каткова (он же и был наиболее пострадавшим).

Сначала исправление ошибок шло в рабочем режиме: как только она обнаруживалась, так тут же исправлялась и шла в рабочую версию. Скоро выяснилось неудобство такого режима — якобы исправление ошибки приводило к неработоспособности транслятора. После этого был введен режим версионности: исправление ошибки обкатывалось на потоке тестов и задач, и только потом это исправление вносилось в рабочую версию транслятора. Хранителем версий и официальным (и единственным) исправителем был Юра Михалевич.

При отсутствии алфавитно-цифрового печатающего устройства вся информация об ошибках печаталась во внутренних цифровых кодировках транслятора. Возникла необходимость в посредничестве между транслятором и пользователями — перевода кодировок на “человеческий” язык и информировании пользователей о месте и характере ошибок. Такое посредничество осуществляла Альфа-группа, в которой главные роли играли Володя Минаев, Лиля Корнева, Маша Легостаева.

Отлаживать Альфа-программы было трудно — большая оптимизация не давала возможности хорошо установить соответствие между машинной программой и исходным тестом. Нужен был специальный инструмент. Такой инструмент — Альфа-отладчик — был создан Михалевичем. На основании указаний к отладке (какую историю исполнения надо выдать) он делал нужные вставки в терминах Альфа-языка в исходный текст. Отредактированный таким образом текст потом и транслировался.

Конечно, Альфа-система с точки зрения современных требований к программному продукту была несовершенной: тяжела в использовании, с плохим пользовательским интерфейсом, требовала специальных знатоков-посредников и т.п. Однако благодаря тому, что она давала код весьма близкий по эффективности к ручному программированию, Альфа-система использовалась везде в тех организациях, где нужна была высокая эффективность программ. Это были сотни организаций по всей стране. Наша Альфа-группа ведала рассылкой новых версий всем пользователям. В некоторых организациях (правда, немногочисленных) Альфа-систему даже модифицировали для своих нужд.

Л. А. Корнева
ИСТОРИЯ АЛЬФА-ГРУППЫ

Состояние вычислительной техники 50–60-х годов

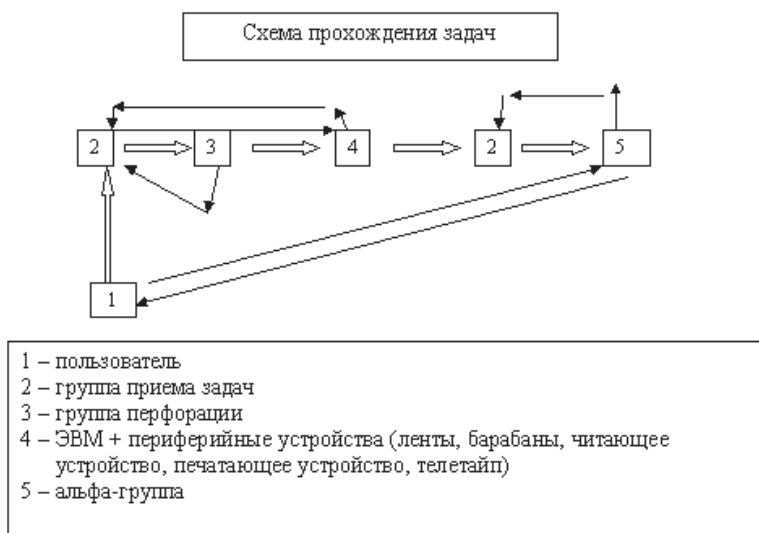
Для того, чтобы понять в сегодняшние дни смысл и назначение Альфа-группы, следует сначала сказать о той вычислительной технике, на которой делался Альфа-транслятор и которая определяла предоставляемый сервис, и о прохождении задач в те времена.

Задача поступала на машину в виде колоды перфокарт, которая ставилась на читающее устройство (ЧУ) и осуществлялся ввод. Через несколько запусков перфокарты “махрились” и ЧУ “заедало” их. Иногда из-за неисправности ЧУ такое происходило и с новой колодой. Помимо порчи перфокарт, ЧУ иногда просто неверно считывало код, пробитый на перфокарте. Испорченную перфокарту(ы) можно было либо заново набить, либо отдублировать на специальном устройстве, которое также не гарантировало 100% правильности. Магнитные ленты были не на бобинах, а в карманах стоек магнитофонов и перематывались из одного кармана в другой. Работа магнитных лент и барабанов оставляла желать много лучшего. А Альфа-транслятор находился на магнитной ленте и считывался по блокам. Сбои были настолько частым явлением, что даже в начале 70-х программисты не очень верили в то, что внешние устройства могут проработать хотя бы месяц без сбоя. Сама ЭВМ (сначала М-20, затем М-220, а позднее БЭСМ-6) также не отличалась хорошей работоспособностью, особенно М-20. Печатающее устройство (ПУ) выдавало результат в виде рулона узкой бумажной ленты, где нецифровая информация печаталась в восьмеричном виде. Кроме того, возможна была выдача на телетайп, которым пользовались те, кто считал прогноз погоды.

Система прохождения задач

Хотя и не сразу, но была принята следующая схема прохождения задач пользователя.

Задача в рукописном виде (если она сдавалась в первый раз) посту-



пала в группу приема задач, руководителем которой было Анна Сидоровна Мучная, известная всем программистам того времени. Из группы приема задача поступала либо в группу перфорации (если надо было набить задачу или исправить отдельные перфокарты), либо непосредственно в группу операторов ЭВМ. После пропуска задачи на ЭВМ она поступала снова в группу приема задач, а затем задачи, написанные на входном языке АЛЬФА или АЛГО.Ле 60, поступали в Альфа-группу (если выдачи содержали не результат). При любой ошибке или модификации задачи весь цикл повторялся. Обычно, если ЭВМ работала нормально, то задачу можно было пропустить два раза за день, но очень часто это было возможно только один раз, а то и еще реже. Пропускная способность машины была очень далека от требуемой.

Каждый пользователь должен был пройти всю эту цепочку, при этом сбои могли произойти в любом звене этой цепочки. Случались (но к счастью, не очень часто) такие “везучие” бедолаги, которые наткнулись на сбой в каждом звене до получения результата. Иногда пользователи сами перфорировали, пропускали свои задачи и сами приходили в Альфа-группу. Но время на ЭВМ распределялось в группе приема задач.

Следует отметить, что к этой цепочке прохождения задач добавля-

ется Альфа-транслятор, состоящий из 30 блоков (в восьмеричной системе), и через каждый из этих блоков должна была проходить задача. Требуется не так уж много воображения, чтобы представить себе сколько иногда сил и нервов стоило программисту довести программу до требуемого результата. Надо было пройти через собственные ошибки, сбои, которые могли произойти в любом звене цепочки прохождения задач, и через все блоки транслятора!

“Сервис” Альфа-транслятора

Сообщения Альфа-транслятора представляли собой текст, но выдавались в восьмеричном виде пока не появилось АЦПУ. Поскольку не было привязки сообщения к исходному тексту программы, то приходилось по внутренней кодировке определять текст и привязывать этот текст к тексту исходной задачи. Внутренние выдачи Альфа-транслятора, к которым приходилось часто прибегать для обнаружения ошибок после 3-го блока, имели несколько внутренних кодировок. Все блоки могли выдавать весь текст задачи во внутренних кодировках. Для обнаружения нетривиальной ошибки умозрительно определялось, примерно с какого блока необходимо выдавать отладочные выдачи. Требовалось просмотреть первую отладочную выдачу, сопоставить с искомым фрагментом исходного текста, а затем последовательно после каждого блока отыскивать этот фрагмент и проверять правильность его обработки соответствующим блоком. Величина фрагмента зависела от самой ошибки. Причин ошибок было множество: ошибка программиста, ошибка при перфорации, перепутанные перфокарты, сбой машины (процессора или периферийных устройств). Транслятор был написан в коде машины М-20 и, кроме описания входного языка АЛФА, не имел никакой документации. Оптимизации, применяемые в трансляторе, по-моему, впервые осуществлялись так комплексно, в таком количестве. Благодаря этим оптимизациям транслятор генерировал очень хороший код (насколько мне известно, по качеству исполняемых программ ни один транслятор в мире до сих пор не превзошел Альфа-транслятора). Но обратная сторона этой оптимизации состояла в том, что не было линейного соответствия между исходным текстом программы и текстом программы в коде машины, что естественно осложняло поиск ошибок. Когда был неверный счет, то выдавалась исполняемая программа в коде машины, которую надо было просмотреть и сопоставить его исходному тексту. Таким образом проверялось соответствие того, что написа-

но, тому, что задумано. Если ошибка обнаруживалась у программиста, то поиск ошибки на этом заканчивался. В противном случае надо было делать отладочные выдачи транслятора. Далеко не всегда ошибка программиста или транслятора была очевидной, т.е. обработка текста программы осуществлялась неправильно из-за того, что где-то в другом месте была ошибка у программиста. Заранее этого никто сказать не мог. Понятно, что при таком “сервисе” Альфа-транслятора без помощи разработчиков или сопровождающих этот транслятор программисту было не обойтись.

Создание и назначение Альфа-группы

В начале декабря 1963 г. было принято решение: с 10 января 1964 г. начать прием задач от внешних организаций. С 16 по 20 декабря 1963г. были организованы курсы по Альфа-транслятору, которые вела Р. Д. Мишкович.

Было принято также решение от лабораторий ВЦ выделить по 1–2 сотрудника для обучения Альфа-транслятору в отделе программирования до 1 февраля 1964 г.

С 9 декабря 1963 г. В. П. Минаев был назначен старшим оператором Альфа-транслятора, именно с него началась Альфа-группа. Вторым сотрудником Альфа-группы была М. Н. Легостаева, третьим сотрудником с сентября 1964 г. была я. Всего в Альфа-группе за время ее существования работало 12 человек. Но одновременно в ней работало самое большее 6 человек. Постоянными оставались только Маша Легостаева и я. Из Альфа-группы я ушла в 1970 г. После чего группа была реорганизована, и ее дальнейшая судьба мне не известна.

В Альфа-группе достаточно долго проработала Большакова Нина, через Альфа-группу прошли Долгая Лариса, Милютин Юра, Шцурик Слава, Лукинцов Саша, Грищенко Галя, две Тани — Безьева и Тропникова.

Альфа-группа была создана для стимулирования широкого использования Альфа-транслятора и обеспечения консультациями по работе с ним и по входным языкам Альфа, Алголу. Это была первая группа, которая служила посредником между разработчиками и программистами. Опыта не было ни у кого, все начинали с нуля. Позднее по образцу нашей группы была создана подобная группа в ВЦ АН и в Петергофе у Н. Н. Лозинского. Вероятно, подобные группы создавались во многих других организациях, которые брали Альфа-транслятор, но о них мне ничего не известно, тогда как с московской Альфа-группой и с группой

Лозинского мы поддерживали добрые отношения.

В функции Альфа-группы входили консультации по входным языкам Альфа и Алголу и идентификация ошибок с локализацией их места в исходном тексте программы. Если ошибка в исходном тексте была не найдена, то к нахождению ошибки привлекались разработчики. Основным помощником Альфа-группы из разработчиков был Г. И. Кожухин. Он мог всегда определить, действительно ли это ошибка транслятора и, если да, то кому из разработчиков ее переадресовать. Частыми помощниками Альфа-группы были также Ю. М. Волошин и И. В. Поттосин. Все эти люди активно способствовали становлению Альфа-группы и в большой степени определили ее лицо. Я думаю, что без этих людей Альфа-группа никогда не получила бы того признания и уважения, которым она пользовалась не только в Новосибирске. Помощь этих людей была неоценимой не только для Альфа-группы, но и для любого программиста, который мог всегда рассчитывать на квалифицированную консультацию и ценный совет.

Из-за плохой пропускной способности машин и постоянных сбоев где-нибудь всегда была масса недовольных программистов, особенно тех, кому сильно “везло”. Кроме того, все программисты без исключения были уверены в том, что у них в задаче ошибок нет, а если задача выдана с ошибкой, то эта ошибка не его. Особенно это было характерно для тех случаев, когда задача хотя бы один раз выдавала правильный результат. Довольно часто ошибки возникали при модификации задачи. Но программист утверждал, что у него все правильно, задача считала, он НИЧЕГО не менял. Тогда приходилось спрашивать, а для чего он пропускал задачу еще раз. И выяснялось, что он все-таки что-то менял и именно в этих изменениях надо искать ошибку или по крайней мере тщательно проверить изменения. Это могли быть измененные входные данные или, по мнению программиста, “пустяковые” изменения в программе, которые не могли привести к ошибке. Довольно часто он, к сожалению, заблуждался. Но первой его реакцией на получение задачи с ошибкой был возмущение. И первыми, кто им попадал под руку, была Альфа-группа, которая принимала на себя эти удары. Все первые претензии и раздражение программистов находили выход именно там и очень часто никуда дальше не шли. Конечно, их можно было понять, поскольку претензии часто оказывались справедливыми, но не имели никакого отношения к Альфа-группе. И таких дней в Альфа-группе было гораздо больше, чем мирных и спокойных. Чаще всего

такая “зарядка” была с утра. Несмотря на такое почти постоянное напряжение, редко кто из Альфа-группы позволял себе рассердиться или возмутиться в ответ на незаслуженные обвинения. Сотрудники Альфа-группы действительно по мере сил старались помочь программистам, и это позволяло сохранять доброжелательные отношения. Часто кто-нибудь разряжал обстановку шуткой. Если в Альфа-группе кто-нибудь оказывался не очень занят, а обнаруживалось, что ошибка в программе программиста не по его вине, то сами старались исправить ее и пропустить задачу вне очереди.

Услугами Альфа-группы пользовались программисты всех институтов Академгородка, институтов и почтовых ящиков города Новосибирска. Кроме того, довольно часто приезжали иногородние посчитать свои задачи.

Бывали случаи, когда у командированных срок командировки кончался раньше, чем был получен результат, и иногда эту задачу оставляли в Альфа-группе “доделать”. Однажды вот по такому случаю Минаев Володя сочинил даже песню. Рабочий день уже кончился и Минаев пел эту песню под гитару (где он взял гитару, не знаю!). Слова этой песни забылись, кроме одной строчки “Девки пишут письмецо Ярыгину”. Ярыгин был командированный из Якутии, который как раз оставлял нам задачу, и мы с Машей Легостаевой действительно сочиняли письмо этому Ярыгину под эту песню.

Структура Альфа-группы

В Альфа-группе отсутствовало иерархическое деление сотрудников. Был просто старший в группе, с которого все спрашивали как в самой группе, так и вне ее. Были просто более опытные сотрудники и менее опытные. Работу тоже не делили: просто само собой получалось, что более опытным доставались более сложные случаи. Но учиться всему старались все, хотя и в разной степени.

Следует отметить, что не только разработчики Альфа-транслятора оказывали помощь Альфа-группе, но и очень многие сотрудники ВЦ, начиная с дирекции, с пониманием относились к проблемам и нуждам Альфа-группы и не отказывали в помощи. Эта помощь заключалась как в организационных мерах, так и в консультациях, не связанных с Альфа-транслятором. По крайней мере, я не могу припомнить случая, чтобы мне отказали в помощи, когда я за ней обращалась.

Поскольку никогда не было уверенности, что ошибка, которая бы-

ла выдана Альфа-транслятором является ошибкой программиста, которую он сам должен был искать, Альфа-группе приходилось докапываться до сути ошибки, чтобы исключить ошибку транслятора, т.е. по существу отлаживать чужую программу. Это, в конечном счете, способствовало тому, что некоторые программисты стали считать обязанностью Альфа-группы отлаживать их программы, а им достаточно только написать их.

Доброжелательность и желание сотрудников Альфа-группы облегчить программисту общение с Альфа-транслятором и ускорить получение результата тоже иногда способствовали потребительскому отношению некоторых программистов к Альфа-группе. Правда, у откровенно нахальных я старалась пресечь такие поползновения, но не могу поручиться, что другие сотрудники могли делать то же самое.

Помнится, был случай, когда у одного из программистов была ошибка в задаче, и я ему объяснила (и даже не один раз), как и что нужно было сделать. Однако он упорно сидел и ждал. Я спросила его, чего он ждет, а он ответил, чтобы я за него все это написала. На что я сказала, что готова объяснить ему еще раз, но делать за него не буду. А он, улыбаясь, говорит: “Я возьму вас измором.” Это был явный случай потребительского отношения. Были, конечно, случаи не такие явные, но такого же отношения к работе Альфа-группы. Но следует сказать, что жизнь нам это не портило. Если время позволяло (т.е. не было каких-то срочных дел), то мы помогали и таким программистам в надежде, что они, в конечном счете, сами научатся отлаживать свои собственные задачи.

Приходилось иногда слышать из третьих рук такое мнение программистов: “Иди в Альфа-группу, там тебе все скажут и сделают.” С одной стороны, это означало, что сотрудники Альфа-группы обладали достаточной квалификацией, чтобы помочь программисту решить его проблемы; но, с другой стороны, у не очень радивых программистов порождало желание переложить часть своей работы на Альфа-группу.

Особенно это было характерно для второго периода эксплуатации Альфа-транслятора, когда увеличилось число профессиональных программистов, в отличие от того времени, когда программированием занимались специалисты из различных областей (физики, химики, механики, прикладные математики и т.д.), причем достаточно высокого уровня. Во втором случае был интерес к познанию новой для них области, а в первом случае была чисто профессиональная работа программиста.

Учителя и помощники Альфа-группы

Говоря об Альфа-группе, больше всего хочется говорить о людях, поскольку любые идеи реализуют люди и от них зависит, как воплощаются эти идеи. В первую очередь, конечно, хочется отметить Г. И. Кожухина, Ю. М. Волошина, И. В. Поттосина, Б. А. Загацкого, Т. А. Темноеву, которые всегда и безотказно оказывали любую помощь, были главными вдохновителями и учителями группы.

Следует отметить доброжелательность отношений как внутри Альфа-группы, так и между Альфа-группой и разработчиками и программистами. Доброжелательность разработчиков, их постоянная готовность помочь и научить, их терпение определяли и ответную реакцию Альфа-группы научиться и постепенно снять бремя сопровождения с разработчиков. Первые пользователи были не профессиональные программисты, а специалисты в самых разных областях, которые взялись осваивать смежную профессию, были люди очень интересные, активные. Общение с ними было очень приятным и интересным. Почти все были молоды и обладали большим чувством юмора.

Юмора иногда, видимо, было с избытком. Какое-то время Альфа-группа располагалась через стенку с кабинетом А. П. Ершова. Комната Альфа-группы почти всегда была полна народу, было шумно и часто раздавался смех. Разработчики также сидели в соседних комнатах и иногда заходили к нам в комнату просто расслабиться и пообщаться с пользователями. Конечно, этот шум мешал Андрею Петровичу, и в один прекрасный момент он не выдержал, и нас переселили в другое место, подальше от разработчиков.

Первые программисты на языке Альфа

Хочется сказать еще о первых программистах, пользователях Альфа-транслятора. Это были, как правило, умные люди, квалифицированные специалисты в своей области, которые решали свои задачи. С ними было легко и интересно. Легко потому, что им самим хотелось все понять и узнать и они быстро все схватывали. Интересно потому, что первые — почти всегда энтузиасты, они активны, много знают и обладают хорошим чувством юмора и потому не столь раздражительны. Общение с этими людьми доставляло большое удовольствие не только нам, но и разработчикам, которым было интересно узнать, какого типа задачи решают пользователи, какие у них возникают проблемы. У нас у всех

остались очень теплые воспоминания о тех людях.

Помнится, что где-то в конце 60-х мы с Г. И. Бабецким оказались в Алма-Ате в командировке (ставили Альфа-транслятор геологам). Бабецкий знал, что там теперь живет наша одна из первых и активных пользователей из города Новосибирска Гапоненко. И он ей позвонил. Она так обрадовалась нам, устроила специально семинар в ИЯФе, а потом — дома прием в нашу честь. Когда я была в командировке в Петергофе у Н. Н. Лозинского, меня пригласили на их внутренний праздник, позаботились о моем ночлеге в Петергофе, т.к. я остановилась в гостинице в Ленинграде. Надо сказать, что такое радушие и доброжелательность мне приходилось встречать везде, откуда к нам приезжали пользователи Альфа-транслятора, несмотря на непродолжительное знакомство в Новосибирске.

Когда-то мы с Г. И. Кожухиным и И. В. Поттосным мечтали проехать по всем столицам республик и поставить там Альфа-транслятор. Но после одного вояжа Киев—Минск—Ленинград—Москва, из-за проблем с гостиницами в те времена, эта мечта умерла, да и дел дома было много. Пропало желание куда-то ездить, и предпочитали принимать у себя.

Распространение Альфа-транслятора

Альфа-транслятор распространялся на перфокартах и занимал около 4000 перфокарт. Это был достаточно приличный объем и вес. Каждый экземпляр транслятора получался на устройстве дублирования, что занимало не один час времени. Специальной службы по получению экземпляра Альфа-транслятора не было, и из положения выходили по-разному. Поэтому мы старались договариваться с теми, кто брал транслятор, что в своем городе они будут распространять транслятор, объясняя наше трудное положение, но как правило, этот договор не соблюдался. Особенно это было характерно для московских организаций. Люди ехали в Новосибирск вместо того, чтобы взять транслятор под боком. Иногда это объясняли трудностью получения информации. Скорее всего не было той доброжелательности, которую они встречали в Новосибирске. В какие-то организации мы ездили сами ставить транслятор, из каких-то приезжали представители и забирали транслятор. Не во всех организациях он использовался одинаково интенсивно. Вероятно там, где смогли организовать более или менее квалифицированную Альфа-группу, транслятор использовался больше. По моим учтенным

пометкам мы поставили около 50 экземпляров Альфа-транслятора. С некоторыми организациями устанавливались длительные связи, некоторые организации пропадали сразу же после того, как получали транслятор.

Не так давно встречались первые сотрудники Альфа-группы, естественно вспоминали старое и говорили: “Какие были замечательные люди! Какие все-таки были замечательные времена! Несмотря ни на что!”

Из воспоминаний разработчиков

1. Вместо оператора на машине оказался Ю. М. Волошин. Приходит программист и уверенно ставит на ЧУ колоду перфокарт, не снимая “рубашки”, и говорит: “Запускайте!”. Волошин посмотрел на программиста ...и запустил. ЧУ работает, а колода на месте. Программист спрашивает удивленно-возмущенно, почему не вводится колода. Волошин отвечает: “А Вы бы еще ее в портфеле поставили!”. Программист очень оскорбился.

2. Резервированные служебные слова в Альфа-трансляторе разрешалось писать по-английски и по-русски и использовать любое доступное устройство для набивки перфокарт. Все новосибирцы использовали как правило русский язык, а приезжие, особенно из Москвы, английский. (Московские разработчики трансляторов с АЛГОЛа утверждали, что надо выгонять тех программистов, которые не знают английского. Правда, позднее они изменили свою точку зрения, но, по-моему, несколько поздно). Ю. М. Волошин приходит в Альфа-группу и видит задачу, где служебные слова написаны по-английски. Тут же стоит автор задачи. Волошин у него спрашивает: “Вы говорите по-английски?”. Тот испуганно отвечает: “Нет!”.

3. Бабецкий Г. И. собрался покататься на яхте и уже готов был отплыть. Но уву! В этот момент появился гонец на велосипеде: “У тебя ошибка!”. Пришлось попрощаться с яхтой.

4. Бабецкий Г. И. (еще в геологии) пришел вечером на машину без пропуска, а вахтер его не пускает. Обошел все здание, нашел открытое окно и влез. Проработал всю ночь и утром, как “белый человек”, идет мимо того же вахтера, а тот удивленно спрашивает: “А ты откуда взялся?”. Вот так раньше иногда ходили на работу!

А. Ф. Рар

ИСТОРИЯ ЭПСИЛОН

Как из Сигмы вылупился Эпсилон

Я пришел в команду Ершова в начале 1965 года, в момент, когда команда эта переживала состояние глубокого удовлетворения после завершения (в основном) первой титанической работы над созданием Альфа-транслятора и представлением его на состоявшейся в Академгородке конференции по трансляторам. Одним из следующих направлений работы отдела была намечена “разработка языков и трансляторов для задач символьной обработки”. А. П. Ершов поручил мне и Л. Л. Змиевской разобраться в том, какие из существовавших к тому времени языков могли оказаться полезными для этой цели. (Результатом этой деятельности стал отчет [1].) Но в то время, когда я разбирался с чужими языками, мне стало известно, что И. В. Поттосин, М. М. Бежанова, Б. А. Загацкий и В. Л. Катков разрабатывают уже некий новый язык Сигма для символьной обработки. В январе 1966 г. я присоединился к этой группе.

Заявленной целью разработчиков языка было предоставить удобный рабочий инструмент системным программистам (тот инструмент, которого так не хватало авторам Альфа-транслятора). Мы стремились достичь разумного компромисса между такими требованиями, как простота и удобство языковых средств, простота трансляции и возможность получения эффективной программы. Работа над языком шла в обстановке оживленных дискуссий (в них нередко участвовал также Г. И. Кожухин), мы принимали решения, отвергали их, вновь возвращались к отвергнутым. В марте мы представили на рассмотрение А. П. Ершова очередной, достаточно продвинутый вариант языка.

Выслушав нас, Андрей Петрович стал размышлять вслух о том, как выношенный нами плод можно было бы дальше усовершенствовать. Идеи стали рождаться на ходу. Открытые процедуры, которые были в нашем проекте, превратились в макросы, а от исходного языка почти ничего не осталось. Зато оказалось, что новый вариант Сигмы есть не что иное, как СИМвольный Генератор и МАКроассемблер, что

неожиданным образом оправдало название языка. Нам было поручено разрабатывать Сигму в духе новых идей.

Как раз в это время “Новый мир” опубликовал “Театральный роман” Булгакова, и я невольно припомнил ту сцену, в которой режиссер Иван Васильевич (шаржированный Станиславский) говорит автору: “Ваша пьеса тоже хорошая, теперь только стоит её сочинить, и всё будет готово.” Эту злободневную цитату я привел, выходя от шефа, Бену Загацкому. Мы посмеялись, но что нам было делать? Решили, что журавля в небе (новую “сигму” — сумму всех прекрасных идей) мы когда еще поймаем, а скромные, маленькие (как “эпсилон” из теории пределов) средства для работы системных программистов нужны уже сейчас. Поэтому, с одной стороны, я вместе с Кожухиным стал разрабатывать ершовские СИГМА-идеи, а с другой стороны, прежняя группа продолжала работать над первоначальным языком, переименованным нами в Эпсилон-язык. (Впрочем, для этого названия была придумана и “расшифровка”: “Элементарный Преобразователь СИМвоЛьнОй иНформации”.) Спустя некоторое время Ершов узнал от меня, что прежнюю работу мы не бросаем, и недовольно произнес: “Как мне не нравятся эти подпольные занятия.” Я постарался об Эпсилон в его присутствии больше не вспоминать, но не всегда это удавалось. Однако спустя некоторое время он сам сказал: “Я обдумал и вижу, что Эпсилон — вещь тоже полезная.” После этого убедительно рассказал, в чем её польза.

К этому времени приводили язык к окончательному виду (особенно с учётом начавшегося уже написания транслятора для М-20) только мы с В. Л. Катковым. Иногда, размышляя над принятием того или иного решения, мы не могли сразу сообразить, как оно стыкуется с решениями, уже ранее принятыми, и должны были перечитывать свои записи. В один из таких моментов я сказал Каткову: “Вот так же и Маркс, наверно, говорил Энгельсу, когда они разрабатывали марксистскую теорию: «А посмотри-ка, Фриц, что мы по этому поводу написали в Манифесте?»” Но пора рассказать вкратце, как выглядел Эпсилон.

Как выглядел Эпсилон

Это был исключительно простой язык. Переменные в нем не имели типа и не описывались, идентификатор переменной просто обозначал содержимое ячейки, отведенной для этой переменной. Константы в языке могли быть либо натуральными числами, либо восьмеричными числами, либо — просто и откровенно — наборами двоичных цифр.

Выражения были только двуместные; арифметические выражения обращались со значениями операндов, как с представлениями натуральных чисел; были еще поразрядные операции, операции сдвига и — что существенно — операция извлечения адреса данного объекта (позволяющая программисту управлять до некоторой степени распределением памяти). Не столь тривиальными объектами языка были так называемые “списки” (на самом деле вовсе не списки, а упакованные массивы) и “слова” (умещающиеся в одном машинном слове последовательности элементов разной длины). Непосредственно с символами и текстами язык дела не имел. Программист мог задать двоичную кодировку для символа или множество двоичных кодировок для класса символов. Существовали переходы на метку, условные и безусловные, а также процедуры, замкнутые и открытые. Законным оператором языка была машинная команда, представленная в восьмеричном виде. Именно в таком виде должны были осуществляться операции ввода-вывода. Используемые программой таблицы хранились как помеченные последовательности машинных команд; совместив в памяти такую последовательность с некоторым описанным списком, программист мог использовать для выбора элемента таблицы механизм выборки элемента списка.

Если говорить о недостатках языка, то главный из них проистек из нашей чрезмерной скромности. Мы почему-то решили, что современные, передовые идеи программистской теории не для нашей “эпсилон-окрестности”, и поэтому не только не прислушались к уже известным идеям Дейкстры о вреде меток и пользе структур, но не вняли даже советам Г. И. Кожухина о желательности явных циклов.

Рождение транслятора для М-220

В начале 1967 года началась работа над транслятором для машины М-20. Алгоритм трансляции был написан на самом языке Эпсилон, а затем, имея этот текст как своего рода эталон, мы вручную преобразовали его (со всей возможной экономией имевшихся в нашем распоряжении 4096 ячеек) в программу на машинном языке. Я программировал фрагменты транслятора, связанные со списками и словами; ввод исходного и вывод результирующего текста запрограммировал В. Л. Катков; всю остальную работу сделал И. В. Поттосин. Стоит вспомнить, как проходило при этом наше взаимодействие с машиной. Она была на ВЦ в единственном экземпляре, часто зависала, время для работы на ней распределялось квотами по 5–10 минут; из-за нехватки времени мы иногда

приходили к машине в надежде, что очередной пользователь недоиспользует свою квоту и можно будет урвать пару минуток. В общем, весело было, но из-за всего этого порой приходилось приходиться на работу ранним утром, а возвращаться домой поздним вечером. И вот однажды моя дочь, тогда второклассница, встретила меня запомнившимися мне словами: “ДАЖЕ ПРИ ЦАРИЗМЕ КАПИТАЛИСТЫ заставляли рабочих работать только 12 часов!”.

К осени 1967 года система была закончена и началась её опытная эксплуатация. В то же время мы стали пропагандировать систему разными средствами; в частности, в ноябре 1968 года на I Всесоюзной конференции по программированию в Киеве я выступил с сообщением о языке и системе [2]; написал я также коротенькую статью о языке для “Энциклопедии кибернетики” [3].

Расширение Эпсилона и его распространение по стране

В том же 1968 году Г. А. Плотникова и Т. К. Панкевич (аспирантка Ершова из Владивостока) начали работу по усовершенствованию системы для машин М-220 и М-222 (включение механизмов отладки, контроля исходного текста, библиотечных процедур). Тогда же существенную роль в дальнейшей судьбе системы Эпсилон для машин типа М-220 сыграло появление системы “Катунь”. Эта система, разработанная В. Л. Катковым при помощи А. Я. Куртукова, содержала набор стандартных процедур для распечатки различных деловых документов, а также препроцессор, включавший в программу на языке Эпсилон тексты востребуемых ею стандартных процедур. Как раз в это время наша экономика возлагала большие надежды на “автоматические системы управления производством” АСУ и АСУП, и поэтому Эпсилон в соединении с “Катунью” “попал в струю”, оказался нужным для многих предприятий. Взаимную адаптацию и тестирование Эпсилон и “Катуни” я проводил на новосибирском Сибсельмаше и барнаульском Тракторном заводе, а затем наступил период внедрения Эпсилон-Катуни в различных регионах страны.

Здесь я хотел бы воздать хвалу эпохе кустарного производства, противопоставив её временам разделения труда, отделения производства от торговли и прочих прогрессивных черт развитого капитализма. Ведь как хорошо было! Какой-нибудь гончар сделает свои горшки и ездит по ярмаркам и по селам, людей видит и себя показывает. А сейчас! Один делает, другой занимается маркетингом, третий транспортиру-

ет, четвёртый продаёт... Скучно. А у нас и того пуще. Создал человек программный продукт, послал его заказчику через e-mail — вот и вся радость. А тридцать лет тому назад я с чемоданом, набитым перфокартами, приезжал на то или иное предприятие, нуждающееся в нашем продукте (а нуждались, в частности, в Алма-Ате, Одессе, Уфе, Арзамасе), ставил систему, читал лекции о языке и его использовании, а заодно зарабатывал деньги. Не для себя (оправдывал только расходы на поездку), а для института. Дело в том, что институт заключал с каждым таким предприятием договор не просто на поставку, а как бы на создание с нуля данного программного продукта с соответствующей, разумеется, полной оплатой.

Эпсилон для других машин, публикации, раскрутка

Вернусь к моменту запуска первой версии Эпсилон-системы. Читал я как-то лекцию об этом новом языке. Подходит ко мне после лекции один из слушателей и говорит: “Я хотел бы написать транслятор с этого языка для машины БЭСМ-6.” Этот человек был Анатолий Евгеньевич Хопёрсков, долгое время работавший после этого в нашем институте, так много сделавший и так безвременно умерший. Транслятор свой Анатолий Евгеньевич завершил в 1970 году. Примерно в то же время в Московском экономико-статистическом институте В. П. Морозов и Л. Я. Семёнова написали Эпсилон-транслятор для машины Минск-22.

Сразу же встал вопрос о выпуске книг по языку и системам Эпсилон. Таких книжек должно было быть две. Первая [4] была написана мной и Катковым как учебник для пользователя. Вторая [5, первоначальный авторский коллектив — Поттосин, Катков, я и Хоперсков] должна была содержать строгое описание языка (в стиле Сообщения об Алголе-60) и описание особенностей его реализаций для машин типа М-220 и для БЭСМ-6. “Изюминкой” книжки явился алгоритм основного блока Эпсилон-транслятора (для М-220), написанный на самом языке Эпсилон. То есть это был тот самый исходный текст, имея который мы писали транслятор, но — разумеется — тщательно отлаженный. Отладка эта состояла в классической раскрутке (bootstrapping): Эпсилон-текст транслятора я пропускал через написанный вручную Эпсилон-транслятор, а получившийся новый транслятор проверял на нормальных Эпсилон-программах. Раскрутка эта дала заодно ответ на занимавший нас с самого начала вопрос: “Насколько программы, полученные путем Эпсилон-трансляции, проигрывают по сравнению с про-

граммами, написанными вручную?” Оказалось, к нашей гордости, что не так сильно. Например, память, занимаемая рассматриваемым основным блоком транслятора, увеличилась при переходе от ручного его варианта к варианту, полученному в результате трансляции, всего в 1,35 раза. К моменту окончания этой работы Морозов и Семенов завершили “минскую” версию системы. Мы решили включить описание этой версии в нашу книгу, а разработчиков её — в список авторов. Но для последнего требовалось согласие издательства (так как договор уже был подписан). Мы объяснили директору издательства: “Так мол и так, надо увеличить список авторов на двух человек.” Директор мгновенно ответил: “Наоборот.” Прежде чем я успел оценить это ошеломительное заявление, он уточнил: “Список авторов поместим на оборот титульного листа.” Но когда книжка вышла из печати, мы не увидели списка ни на обложке, ни на титульном листе, ни на обороте его, а лишь в конце книги среди выходных данных. Так получилось, что книга эта всюду числится безавторной.

Одним из последних действий, связанных с языком Эпсилон, было представление в 1974 году нашего с И. В. Поттосиным доклада на конференцию в Тронхейме (Норвегия). С докладом должен был выступить я, но — не всякая поездка тогда удавалась. Доклад был опубликован в трудах конференции [6]. Во второй половине 70-х годов победное шествие “машин, совместимых с ИВМ”, как мы называли их иностранцам, нанесло смертельный удар нашим отечественным машинам, и этим завершается история языка и систем Эпсилон.

Список литературы

1. **Змиевская Л. Л., Рар А. Ф.** Языки для обработки символьной информации (критический обзор) / Отчет ВЦ СО АН СССР. — Новосибирск, 1966.
2. **Поттосин И. В., Рар А. Ф., Катков В. Л.** ЭПСИЛОН — система автоматизации программирования для задач символьной обработки // Первая Всесоюзная конференция по программированию; Секция Ж: Теория и общие вопросы программирования. — Киев, 1968. — С. 88–108.
3. **ЕПСИЛОН** // Энциклопедія кібернетики. — Київ, 1973. — Т. 1. — С. 336.
4. **ЭПСИЛОН** // Энциклопедия кибернетики. — Киев, 1974. — Т. 2. — С. 586–587
5. **Катков В. Л., Рар А. Ф.** Программирование на языке ЭПСИЛОН

- / Отв. ред. А. П. Ершов; Сиб. отд. АН СССР. — Новосибирск: Наука, 1972.
6. **ЭПСИЛОН** — система автоматизации программирования задач символьной обработки / В. Л. Катков, И. В. Поттосин, А. Ф. Рар и др.; Отв. ред. А. П. Ершов; Сиб. отд. АН СССР. — Новосибирск: Наука, 1972. — 130 с.
7. **Pottosin I. V., Rar A. F.** Application of machine-oriented language EPSILON to software development // In: Machine Oriented Higher Level Languages. — N.-H. Publ. Co., 1974. — P. 417–434.

И. В. Поттосин

ЭПСИЛОН-ИСТОРИЯ

Эпсилон-язык

Итак, середина шестидесятых. Вовсю идет “алголизация” программирования, которая дает в руки простому вычислителю удобный инструмент самому, а не через специального, умудренного знаниями специалиста-программиста, возможность создать программу и общаться с компьютером без посредников. Мы, разработчики системы Альфа, активные участники этого процесса как его создатели. Однако сама система Альфа писалась непосредственно в машинном языке, и мы уже полностью хлебнули проблем, которые возникают, когда большая программная система (а система Альфа была одной из наиболее крупных программных систем, которые были написаны к этому времени) создается на машинном языке. В наши планы входило создание других процессоров, связанных с обработкой символов: систем компьютерной алгебры, систем автоматического построения алгоритмов (иначе говоря, систем синтеза программ) и других подобных систем, связанных с обработкой текстов, формул или программ. На машинном языке их писать не хотелось, а использовать, как друзья-вычислители, Алгол мы, будучи реалистами, не собирались. Конечно, алгоритм для подобных невычислительных задач на Алголе можно было написать, но для кодировки текстов и экономного их размещения в машинных словах Алгол был не приспособлен, его изобразительные средства ориентировались на вычислительные задачи. Мы с завистью смотрели на вычислителей, которых мы же снабдили таким удобным инструментом, как система Альфа, позволяющей им общаться с компьютером на “естественном” языке, и чувствовали, что “сапожник ходит без сапог”.

Именно это побудило нас, разработчиков системы Альфа Майю Бежанову, Бена Загацкого и меня, взяться за создание языка для собственных нужд. Можно было, конечно, придумать какое-либо расширение Алгола (или Альфа-языка), которое включило бы удобные для нас средства, но мы сразу решили пойти другим путем. Мы хотели мыслить в терминах не Алгол-машины, а в терминах машинного языка (программ-

мирования на машинном языке мы хлебнули с лихвой и чувствовали его довольно хорошо): в терминах двоичных кодировок, упаковки информации в машинные слова и пр., но только облекая все в более удобные, отдалившиеся от синтаксиса машинного языка одежды. Иначе говоря, мы от “нашего” языка хотели близкой к машинному языку семантики и удобного синтаксиса. Заметим, что подобная же, по-видимому, идея была и у Никлауса Вирта, когда он создавал ориентированный на архитектуру ИБМ-360 язык ПЛ-360, ставший прототипом большого числа так называемых машинно-ориентированных языков или “языков повыше уровнем” (higher level languages). Разница между нами была в том, что Вирт ориентировался на конкретную архитектуру, а мы — на конкретный класс задач.

Второе, из чего мы исходили при разработке языка — это скорость трансляции при хорошем качестве получаемых программ. Мы не предполагали практически никакой оптимизации и с самого начала считали, что язык будет содержать только такие конструкции, о которых мы заведомо знаем, как их эффективно реализовать. В каком-то смысле это нас толкало на обеднение языка, и он-таки и получился достаточно элементарным. Это нас не смущало, и мы его называли с горделивой скромностью — Эпсилон, что подчеркивало, что это заведомо “маленький” язык. В частности, это сказалось на наборе операторов управления: он состоял только из операторов перехода и операторов условного перехода. Выражения были только бинарными; язык был бестиповый: значение предполагалось двоичным, а как его понимать — как целое число или как битовый вектор — это зависело от операции; структурных значений всего два типа — машинное слово, разбитое на последовательность слогов, возможно различной длины, и вектор одномерных слогов, занимающий последовательность машинных слов и т.д. и т.п.

Формирование языка происходило в жарких спорах. Надо было найти компромисс — пройти между Сциллой недостаточно хорошей выразительности и Харибдой неэффективной реализуемости. Решения принимались, потом отвергались, потом — с изменениями и модификациями — к ним возвращались снова. На наши решения влиял и тот неформальный язык спецификации программ, который мы когда-то с Геней Кожухиным предложили для Альфа-системы. К первоначальному коллективу присоединился Саша Пар с его хорошим чувством языков. Довольно часто мы прибегали к советам Гены Кожухина.

Контуров языка уже вырисовывались, когда об этой работе узнал

Андрей Петрович — до этого вся наша деятельность шла вне всяких планов отдела. Какие-то вещи ему понравились, но, по-видимому, явная элементарность языка и его четкая прагматическая ориентация на уже известные приемы трансляции, а значит, и отсутствие исследовательских проблем его разочаровали. Он стал предлагать идеи развития языка, которые требовали решительного пересмотра всех решений. Если бы контуры языка не сложились, ему с его авторитетом было бы легко нас переломить, однако работа над языком зашла уже слишком далеко, чтобы все менять. Кончилось тем, что он смирился с нашей работой, а свои идеи он стал разрабатывать в другом проекте — проекте языка Сигма, к которому он привлек Сашу Рара. Правда, эффект был еще и в том, что он попросил привлечь к нашей работе Владика Каткова, который как раз собирался переориентироваться от вычислительной математики к системному программированию.

Система Эпсилон-М

Так получилось, что Майя Бежанова и Бен Загацкий занялись другими проблемами, и окончательную версию языка (май 1967 г.) разрабатывали мы троим: Владик Катков, Саша Рар и я. Сразу же приступили к созданию транслятора для М-220. Поскольку реализация языка уже просматривалась при его разработке, на написание транслятора ушло меньше месяца. Я писал основную часть, Саша Рар — реализацию описаний структурных данных и операций над ними, Владик Катков — ввод и вывод. С осени 1967 г. началась опытная эксплуатация системы, в которой активно участвовали Тома Панкевич и Галя Плотникова.

К 1968 году система “вышла на простор”. Она использовалась и в ВЦ СО АН СССР, и в других организациях. С ее помощью были написаны многие системы: и языковые процессоры, и системы компьютерной алгебры, и другие подобные системы. Особую роль она играла как основной инструмент написания программного обеспечения системы коллективного пользования АИСТ-0: все это программное обеспечение размером в несколько сотен тысяч команд — и операционные системы, и трансляторы, и информационные системы, и системы компьютерной алгебры, и игровые программы (кроме ядра ОС, который работал на Минске-22 и был написан в мнемонике Минска) — было написано на Эпсилон.

Сам транслятор был написан тоже на Эпсилон, была осуществлена раскрутка (bootstrapping), благодаря чему все дальнейшее развитие

системы осуществлялось с помощью ее самой.

Итак, система использовалась, доводилась, развивалась. Основную работу по доводке системы, ее сопровождению и распространению выполнял Саша Пар. Он же вместе с Владиком Катковым написал учебник по программированию на языке Эпсилон.

В 1967 году система включала только транслятор. Надо было строить систему — действительно систему — программирования, и в 1968 году за это взялись Галя Плотникова и Тома Панкевич. К концу 1969 года была разработана система Эпсилон-М, она уже включала, помимо немного модернизированного транслятора (был реализован во всей полноте механизм открытых процедур), отладчик (более точно, отладочный редактор, который вставлял в исходный текст операторы, реализующие заданные операции отладки), контроль исходного текста, вставку библиотечных процедур. Все эти работы, как уже говорилось, использовали язык Эпсилон как инструмент.

Другие Эпсилон-системы

В 1968 году в отдел пришел Толя Хоперсков. Он был сотрудником Института гидродинамики и последнее время занимался вычислительной механикой, программировал ряд задач и увлекся программированием как профессией. Со свойственной ему иронией он это отмечал так: “Программист, как минер, ошибается один раз в жизни — когда он выбирает специальность”. В результате такой ошибки он попросился к нам в отдел, в нашу лабораторию, и я предложил ему написать вариант транслятора для БЭСМ-6. Хотя общие алгоритмы трансляции уже были опробованы в системе для М-220, но надо было их трансформировать для БЭСМ-6, а кроме того, доопределить семантику языка применительно к БЭСМ-6. Дело в том, что машинно-независимая семантика Эпсилон только частично покрывала семантику языка (так называемый эталонный уровень семантики), а полная семантика языка описывалась образами: каждой конструкции языка сопоставлялся ее образ в машинном языке. Вот создание таких образов для БЭСМ-6 и было одной из задач Толи. Начав работу в 1969 году, в начале 1970 года он уже завершил разработку и отладку системы Эпсилон-6.

В те же годы сотрудники Московского экономико-статистического института В. Морозов и Л. Семенова создали Эпсилон-систему для Минск-22 (система Эпсилон-Минск), доопределив аналогичным образом семантику Эпсилон для архитектуры Минск-22. Таким образом,

язык Эпсилон в 1970 г. был реализован для всех основных отечественных компьютеров.

Системы Эпсилон успешно эксплуатировались на этих компьютерах, пока они (кроме БЭСМ-6) не были вытеснены “отечественными” ЕС ЭВМ. Где-то году в 1974-м была предпринята попытка разработать систему Эпсилон для ЕС ЭВМ, но неудачный выбор разработчиков не дал осуществиться этому проекту. Язык Эпсилон ушел в прошлое вместе со старыми компьютерами, но сыграл большую и очень полезную роль в создании различных программных систем для этих компьютеров.

А.Ф. Рар
ИСТОРИЯ СИГМЫ

Суть языка Сигма

Сигма определялась как многоуровневая система с некоторым числом свободных параметров. Базисные операции языка не имели своей семантики, имелись лишь стандартные процедуры для адресации объектов, распределения памяти и т.д. Вся связь системы с конкретной машиной осуществлялась через упомянутые свободные параметры. Спецификация параметров переводила язык со стандартного уровня на машинный уровень. Благодаря механизму процедур и макросов основная часть любой программы могла быть написана независимо от машинного кода, так что машинная зависимость программы определялась программированием процедур наинизшего уровня. Стандартные процедуры языка давали возможность создания списочных структур и работы с ними.

Доклад для конференции в Пизе

К середине 1966 года идейная разработка языка Сигма достигла такого уровня, что стало возможным представить эту работу на международной конференции “Языки и технология символьной обработки”, которая должна была состояться в сентябре этого года в городе Пизе. Поехать на конференцию с докладом (А. П. Ершов, А. Ф. Рар, “Сигма — символический генератор и макроассемблер”) предстояло мне.

Другим участником конференции от нашей страны (и главой делегации) предполагался С.С. Лавров. Когда я прилетел в Москву и встретился с Лавровым, то узнал от него, что поездка нашей делегации отменена. “В чём дело? — спросил я Лаврова. — Кто-то из нас двоих оказался слишком плохим?” “Нет, — ответил он, — я оказался слишком хорошим.” (Лавров работал, как известно, в сверхсекретных Подлипках у Королёва.).

Не могу сейчас припомнить, как это получилось, но в промежутке между отменой поездки и началом конференции я успел встретиться

с направлявшимся на ту же конференцию Джоном Маккарти. Я передал ему материалы по нашему докладу и попросил его прочесть текст устного выступления от нашего имени. Он это сделал, доклад был напечатан в трудах конференции, а некоторое время спустя, при очередной встрече с Маккарти, мы услышали от него реакцию одного из участников конференции на это выступление: “Я всегда думал, что русские — это бородатые анархисты. Так оно и оказалось.” (Бородатым был Маккарти, а революционно-анархическими оказались ершовские идеи.)

Первые радости

Работу над первым Сигма-транслятором хорошо описал Г.Г. Степанов. Надо только добавить, что происходила эта деятельность в период расцвета “Факела” и что работали Степанов с Синенкиным по трудовому соглашению с этой, столь популярной тогда, организацией.

В какой-то момент, когда до окончания срока соглашения оставалось всего-ничего, прошел первый тест. Я тут же сообщил об этом Ершову, а затем принес ему все материалы по транслятору. Ершов вынул из шкафа бутылку коньяка (“Коньяк для салютов”, — объяснил он), мы выпили по рюмочке, а затем он вручил мне принесенную мною папку и торжественно сказал: “Я только что получил от «Факела» выполненную им часть работы, теперь надо продолжить её”. О продолжении см. также у Степанова.

Г. Г. Степанов

ИСТОРИЯ СИГМЫ

Создание языка

История Сигмы [1] начинается в 1965 году, когда А. П. Ершов инициировал разработку нового языка программирования для работы с символьной информацией.

История создания языка подробно описана А. Ф. Раром в очерке “История Эпсилон”. Там же отмечается, что название языка — Сигма — неожиданно очень удачно стало соответствовать сути разработанного языка, которую можно описать как “СИмвольный Генератор и МАкро-ассемблер”.

Я подключился к работе над проектом Сигма в декабре 1966 года, будучи студентом 4-го курса математического факультета Новосибирского государственного университета. Одновременно со мной в проект пришел и мой однокурсник Б. Ф. Синенкин. И мы под руководством А. Ф. Рара и при участии Р. Д. Мишкович приступили к реализации языка Сигма на ЭВМ М-20.

Три реализации

Всего в истории языка Сигма было три его реализации: на М-20, на БЭСМ-6 и на самом языке Сигма. Первая, конечно, самая памятная, т.к. это была моя первая работа в области системного программирования (да и вообще первая работа). Вторая была выполнена на лучшей, по моему мнению, отечественной машине БЭСМ-6. Третья опиралась на вторую, была раскручена сама через себя и могла генерировать программы как для БЭСМ-6, так и для СМ-4 и ЕС ЭВМ.

Генерация программ для различных ЭВМ

Последняя возможность (способность генерировать программы для различных ЭВМ) потенциально присутствовала во всех реализациях языка Сигма, т.к. выходной язык системы СИГМА никогда не закреплялся жестко, а описывался на специальном языке [2]. Именно методы,

способы и средства описания выходного языка системы Сигма и являлись темой моей дипломной работы в университете, руководителем которой был А. П. Ершов.

Реализация на М-20

Реализация языка Сигма на М-20 представляла собой двухпросмотровый транслятор и административную систему, обеспечивающую во время работы оттранслированной программы выполнение действий над такими объектами языка СИГМА, как массивы и списки. Программирование административной системы осуществлял А. Ф. Рар. Я выполнял реализацию первого просмотра, включавшего в себя лексический, синтаксический и семантический разборы программы пользователя. Этот же просмотр осуществлял открытую подстановку специально помеченных пользователем процедур. Второй просмотр, осуществлявший генерацию выходной программы по описанию выходного языка, реализовывал Б. Ф. Синенкин.

“Ностальгия” по М-20

Думаю, для современного читателя очень любопытным будет описание условий работы на ЭВМ М-20. Оперативная память машины составляла 4096 45-разрядных слов, поэтому большую часть информации приходилось держать на магнитных барабанах (существовали в то время такие), закачивая ее в память по частям по мере надобности. Такого естественного понятия, как виртуальная память, на М-20 не было, поэтому всю работу по подкачке информации и сбросе ее на барабан приходилось планировать и обрабатывать программно. Сам транслятор и административная система находились “разумеется” на перфокартах.

Программировали непосредственно в кодах команд М-20, собственноручно распределяя память под переменные, константы и другие объекты программы. Хотя писали все-таки на некотором подобии автокода, используя мнемонические имена переменных и кодов команд, которые вручную заменяли на числовые значения перед перфорацией. Но самым забавным представляется мне информация, которую машина выдавала программисту и по которой он должен был отлаживать свою программу. Это была так называемая “узкая цифровая печать”. Каждая строчка на длинной узкой полоске бумаги (которую скручивали в рулоны наподобие туалетной бумаги) состояла из 15-ти восьмерич-

ных цифр, определяющих содержимое одного слова памяти машины. Пользователь мог только в определенном порядке выдать на эту ленту значения интересующих его диапазонов ячеек оперативной памяти, а потом заняться их расшифровкой.

К этому следует добавить, что машинное время было распределено между пользователями, поэтому в день можно было пропустить свою задачу один (редко два) раз. И к запуску все тщательно готовились, т.к. очень обидно бывало терять целый день из-за какой-нибудь нелепой описки. Довольно распространенным приемом отладки программы была так называемая “прокрутка”. Программист ставил себя на место машины и начинал исполнять свою (или чужую) программу, выписывая на доске (или листе бумаги) текущие значения некоторых ячеек памяти. Занятие любопытное, позволявшее к тому же отловить наиболее очевидные ошибки при программировании. Был даже такой случай, когда часа полтора мы с А. П. Ершовым “прокручивали” мою программу.

Тем не менее работа по реализации шла довольно успешно. Первый тест прошел 30 октября 1967 года, а к июню 1968, т.е. к концу нашего с Б. Ф. Синенкиным обучения в университете, мы пропускали уже вполне приличные программы. Ну а после окончания университета меня забрали в армию, поэтому работа по реализации языка Сигма на М-20 для меня закончилась.

Реализация на БЭСМ-6

После армии я поступил на работу в Вычислительный центр Новосибирского отделения Академии наук СССР. К этому времени сменилось поколение машин, все большее распространение стала получать БЭСМ-6. Поэтому мы приступили ко второй реализации языка Сигма [3,4]. Общая структура проекта была примерно такой же, а вот условия разработки существенно изменились. Память на БЭСМ-6 была побольше, магнитные барабаны сменились магнитными дисками, для выдачи уже использовались алфавитно-цифровые устройства (АЦПУ). И хотя для ввода по-прежнему использовались перфокарты, программировали мы уже не в кодах команд, а на автокоде БЕМШ. Данная реализация была доведена до работающего состояния, позволившего начать третью реализацию языка Сигма уже на самом языке Сигма [5].

Сигма на Сигма

Последняя реализация выполнялась уже только мной. Основной машиной была по-прежнему БЭСМ-6, а в качестве выходных машин, кроме БЭСМ-6, описывались СМ-4 и ЕС ЭВМ. Со временем на БЭСМ-6 появились терминалы (чаще всего применялись черно-зеленые мониторы марки “Videoton”), перфокарты отошли в прошлое (а так удобно было иметь в кармане несколько штук для различных записей).

Сам язык Сигма [6,7] несколько расширился, в основном за счет добавления новых служебных и стандартных макросов, но его основные концепции остались неизменными.

Широкого распространения система Сигма [8] не получила, но тем не менее с ее помощью был разработан ряд прикладных программ, среди которых, в первую очередь, следует упомянуть программу управления некоторой справочной базой данных, содержавшей адресно-телефонные данные более чем 1500 отечественных программистов. Распечатка данной базы сохранилась, и ею пользуются и сейчас, хотя большинство данных в ней, конечно же, устарело. В конце 80-х годов закончилась эра машины БЭСМ-6, а с ней ушел в историю и язык Сигма.

Литература

1. **Ershov A.P., Rar A.F.** SYGMA — a Symbolic Generator and Macro-assembler // IFIP Working Conference. — Pisa, 1966. — 39 p.
2. **Степанов Г.Г.** Методика описания вычислительных машин для настраиваемых макроассемблеров // Тр. Всесоюзного семинара по вопросам макрогенерации. — Тбилиси, 1975. — 19 с.
3. **Stepanov G.G.** SIGMA — a machine-oriented higher-level language // Machine Oriented Higher Level Language. — N.-H.Publ.Co., Amsterdam, 1976. — 10 p.
4. **Степанов Г.Г.** Реализация системы СИГМА // Теория и практика программного обеспечения (Тр. советско-французского симпозиума). ВЦ СО АН СССР. — Новосибирск, 1981. — 10 с.
5. **Степанов Г.Г.** Опыт использования языка СИГМА // Языки и системы программирования. ВЦ СО АН СССР. — Новосибирск, 1981. — 8 с.
6. **Ершов А.П., Степанов Г.Г.** Машинно-ориентированный алгоритмический язык СИГМА для написания переносимых программ (общее описание). — Новосибирск, 1981. — 30 с. (Препр. / ВЦ СО АН СССР; № 304)

7. **Степанов Г.Г.** Система программирования СИГМА. — Новосибирск, 1982. — 29 с. — (Препр. / ВЦ СО АН СССР; № 391)
8. **Степанов Г.Г.** Пути обеспечения переносимости программ и опыт использования системы СИГМА // Трансляция и преобразование программ. — ВЦ СО АН СССР. — Новосибирск, 1984. — 9 с.

А. А. Берс, А. Ф. Рар
ДИАЛОГ ОБ АЛГОЛЕ 68

(Авторы просматривают архивные материалы
и непринужденно беседуют)

Берс: Так когда мы занялись Алголом 68?

Рар: Я занялся Алголом еще в 66 году, когда Ершов привез, не помню с какой конференции, известие, что наступил следующий шаг в развитии Алгола 60, что называется новый язык “Алгол 67”, и прочитал нам всем лекцию, что это за язык и что там содержится.

Берс: Там уже была двухэтажная грамматика. Какое впечатление на тебя произвело это известие?

Рар: Я был молод и послушен. Ну, язык как язык, дальнейшее развитие и т.д. В чем дело? Всё удовольствие от логической стройности языка и все связанные с этим затруднения я оценил значительно позднее, когда стал писать реальные программы на Алголе 68. В 66-м же году сразу началась работа по созданию русской версии нового Алгола, невзирая на то, что он еще не был доделан. В этой работе были задействованы в основном мы с Андреем Петровичем в Новосибирске и Святослав Сергеевич Лавров в Питере. Между нами началась оживленнейшая переписка; эти материалы есть. Главным образом по поводу составления словаря.

Берс: Это очень интересно. Хорошо бы выдержки из этих писем посмотреть, вспомнить, как какие термины переводились.

Рар: Да нет, они, в основном, мелочные — типа, этот термин лучше перевести вот так...

Берс: А словарь того времени более-менее полный сохранился? Это на самом деле интересно. Потому что я-то в 68 году впал в вашу команду и на самом деле достаточно быстро влип в это дело...

Рар: “Словарь терминов Алгола 67. (Согласован 5 февраля 1967 года.)” Я не помню, с кем он был согласован, может быть, было согласовано у нас с Ершовым.

Берс: Если ты переписывался с Свет Сергеевичем... Вот тут “row” переведено как “ряд”. А вообще-то как-то мало материалов.

Рар: Мало.

Берс: Они что, потом появились? Потому что к тому моменту, когда я в 68 году этим занялся, их было уже довольно много ... Я-то пришел, когда только появился документ амстердамского Математического центра под номером 93...

Рар: MR-93.

Берс: Да. И вот там уже были пресловутые “КВАЗИЛЕДУСЫ” и “КВАЗИЛЕМОДУСЫ” — метапонятия, заканчивающиеся в английском на “ЕТУ”, а в русском варианте начинающиеся с “КВАЗИ” и дающие в числе своих порождений “ПУСТО” ...

Рар: А вот здесь какие-то “clarifications”, “уточнения”. А чьи они и для кого, я не помню. Может они принадлежали Лаврову, потому что у меня здесь замечания по поводу “уточнений” Лаврова. Все они тут какие-то разрозненные...

А в 67 году началась у меня переписка с ван Вейнгаарденом. Вот здесь мое письмо к нему от апреля 1967 года: “Дальнейшее изучение последнего присланного Вами документа по Алголу 67 заставляет меня послать Вам это второе моё письмо. Оба письма не касаются каких-либо принципиальных вопросов и не содержат каких-либо существенных предложений по поводу улучшения языка. Я хочу лишь привлечь Ваше внимание к некоторым сомнительным пунктам как большой, так и малой важности”... после чего следуют “major remarks” и “minor remarks”. И началась такого сорта переписка, так сказать “взаимопроникновение”.

Берс: Какова же была позиция ван Вейнгаардена? И как происходили контакты. Ты ездил куда-нибудь?

Рар: Это было потом. Ездил я в 1970 году в Мюнхен. А до того я бывал только в Ленинграде, где в основном общался с Цейтиным, который к тому времени тоже включился в эту работу. И вместе с Цейтиным мы писали письма ван Вейнгаардену.

Берс: И повлияло ли что из этого на язык?

Рар: Думаю, что если и повлияло, то по мелочам. Скорее, мы выполняли приятную роль критика со стороны. Ну не в этом суть. И попутно мы переводили. В это время ты и включился в это дело.

Берс: Ну, я пришел осенью 68-го. А в конце 68—начале 69-го Андрей Петрович нарисовал громадную такую синтаксическую карту. Это было первое мое впечатление. Да, а участие Андрея Петровича во всем этом до 68 года? Ведь на международные конференции ездил он.

Рар: Его участие было непосредственным в том смысле, что все, что я предлагал по словарю, шло через Андрея Петровича. Да, в начале 68-

го состоялась уже зимняя лыжная школа по Алголу 68 в Бакуриани. Туда ездили я, Ершов, Змиевская и покойный Саша Куртуков. Там были Лавров, Шура-Бура, Любимский, Тыгу. От хозяев были Деканосидзе, Мурман Цуладзе. Было очень хорошо и приятно. Мы читали лекции по Алголу 68 по главам. Ну и кроме того, говорили и о других работах. Я, кстати, об Эпсилон там рассказывал. Энн Тыгу докладывал о своих проектах. Да, еще Брудно там был с едкими критическими замечаниями.

Берс: А, кстати, критика Алгола была достаточно жесткой? И за счет чего? Что в языке ругали?

Пар: Я думаю, ругали за сложность.

Берс: Ну это понятно. Потом появился MR-93, а следующий существенный документ был, как я помню, MR-95. Значит, шел перевод и, в основном, составление словаря. Ну, а что на меня произвело впечатление? Во-первых, синтаксическая карта, которую рисовал Андрей Петрович. Она была в основном древовидная, что мне не очень нравилось из-за большого числа повторных листьев. Потом я помню, что влез во все эти “portmanteaux”, метапонятия, соединяющие в себе разные смыслы — “VIRACT” как соединение “virtual” и “actual”... Если помнишь, я по их поводу заводился, от разных НЕПРОБЕСПАР да еще НИРХНИЙ, и потихоньку начал предлагать какие-то способы избавиться от них. Но к некоторому моменту мы это утрясли, и перевод ушел в “Кибернетику” уже безо всяких там КВАЗИЛЕДУСОВ и КВАЗИЛЕМОДУСОВ... Зимой, в конце 68, мы все ездили на ВКП-1 в Киев и там договорились, что будет издаваться билистинг в журнале у Глушкова.

А в это время у меня появился Валера Грушецкий, и это уже, если не ошибаюсь, 69-й год. Мы с Грушецким придумали, как можно уменьшить количество линий на синтаксической карте — за счет того, чтобы устроить некоторый коллектор. Это опубликовано даже. Чтобы по коллектору сохранялось одно и то же направление. Эту карту мы с Грушецким и выкладывали на полу у меня дома из спичек и фишек. Конкурентов в этом направлении у нас фактически не было. Я даже помню, как я уходил, вернулся, лежит записка “Я все сложил, свет не гашу, жалко проделанных трудов”. А мы в этот момент выкладывали пятую главу, про форматы. Пользовались мы ссылками, ссылки оказались неверными, в результате у нас получился цикл, который было очень трудно расположить. Потом пришли исправления, вся эта конструкция цикла рассыпалась, в последующем пришлось заново приво-

дить в порядок.

В общем, этот перевод занял у нас достаточно много времени, мы практически этим все время и занимались. Хотя, впрочем, и не только этим. Ведь у тебя был Эпсилон, у меня были потоковые схемы, которые потом, в 72 году, Андрей Петрович выставил на симпозиум по теории программирования. Но в 69 году мы уже с весны начали застревать подолгу в Киеве.

Рар: Сперва ты поехал в Киев один, я не смог.

Берс: Потом мы ездили с Грушецким и с тобой. Потом я еще ездил. И на самом деле мы достаточно хорошо продвигались, хотя были фундаментальные трудности из-за того, что нам для издания на двух языках требовалось разнообразие шрифтов. Ван Вейнгаардену было хорошо, у него была ИВМовская пишущая машинка со сменными головками, и он, по-видимому, не ленился менять эти головки. Ведь все метапонятия печатались крупным шрифтом, протопонятия — мелким. Затем курсив, полужирный, — а в целом было по 6 начертаний шрифтов для каждого из текстов. Но когда мы появились в типографии издательства “Наукова Думка”, то выяснилось, что для латиницы того варианта, который мы хотели использовать, нет.

Короче говоря, нам пришлось выбрать комплект двух “разных” гарнитур — “обыкновенная новая” и “литературная”, которые, вообще-то говоря, на нормальный взгляд ничем не отличаются. И плюс какая-то “рубленая” у нас все-таки была. Всё это набиралось монотипом, мы собственноручно ковырялись в тексте прямо на талере шилом, и я ругался сильно по поводу того, что в некотором месте должна быть прямая запятая, а стоит курсивная. В ответ мне, естественно, говорили, что этого никто не увидит, а я возражал: я же увидел. И на самом деле, это различие было существенным. Тексты примеров на Алголе 68 набирались курсивом, и все запятые, которые были там, тоже должны были набираться курсивом; далее шла прямая запятая, и начинался следующий пример. Конечно, все это было очень здорово, но зато приводило к большим техническим затруднениям и огромному количеству опечаток.

Вообще, должен заметить, что среди прочих впечатлений об Алголе 68 на меня произвел приятное впечатление опыт обмена сообщениями об опечатках. Он был очень хорошо спроектирован, достаточно прост и в то же время достаточно точен, позволял описывать любые текстуальные изменения. За всем этими изменениями следили в Москве, Новосибирске, Ленинграде, Брюсселе, Амстердаме, Эдмонтоне. Алгол 68 был

действительно международным проектом, и действовала международная Рабочая Группа. Один раз эта группа, как ты помнишь, заседала здесь, и мы были наблюдателями.

Официальное сообщение о языке было опубликовано в “Numerische Mathematik”, № 14 — последнем номере за 1969 год, и мы уложились в сроки и опубликовали наш перевод в “Кибернетике” в № 6 за 1969 (вышел в январе 70) и в № 1 за 1970, правда, первый номер за 70-й год вышел в июле. К тому времени мы с Грушецким сделали всё-таки эти самые синтаксические карты. Это печатный лист формата А0, у которого с одной стороны была русская синтаксическая карта, с другой — английская. В оригинале это представляло собой четыре ватманских листа. Я помню, как перед ВКП-2 привезли отпечатанные на меловой бумаги кусочки карт. То есть синтаксические правила были напечатаны отдельно, мы их клеили, страшно торопились и были вознаграждены. Потому что эти четыре листа синтаксической карты, наклеенные и расчерченные, мы принесли в гостиницу, в номер к ван Вейнгаардену. И ему ничего не оставалось, как положить их на ковер, на пол, после чего, чтобы читать, ван Вейнгаардену пришлось опуститься на колени, а Валера Грушецкий потом рвал на себе волосы, потому что он забыл фотоаппарат. Я думаю, что фигура ван Вейнгаардена, улыбающегося, стоящего на коленях перед картами и их рассматривающего — вот это и была вполне достаточная награда за проделанную работу.

Пар: Минуточку. Это когда было?

Берс: Это было на ВКП-2, 1970 год. Ван Вейнгаарден делал доклад на ВКП-2, показывал, как при помощи двухэтажной грамматики можно просто проводить вычисления. Показывал универсальность грамматики.

Пар: А разве он с этим докладом не в Ургенче выступал, на конференции в честь Аль-Хорезми?

Берс: Нет, он выступал на ВКП-2.

Пар: Это был его единственный визит..

Берс: Нет, не единственный, по той простой причине, что был он и на Рабочей Группе, которая проходила в Новосибирска. Это был второй его визит. И, по-моему, была еще какая-то конференция.

Пар: Я помню, когда он первый раз приезжал, я его встречал в Толмачево, он привез тогда книгу “Неформальное введение”, которую мы тоже издавали. Её переводил С. Я. Лифшиц.

Берс: Правильно.

Рар: Да, но если вернуться к нашим киевским поездкам...

Берс: Если вернуться, то в июне 70-го и вышел, как я сказал, первый номер за 70 год, в котором, как известно, пять страниц было о чем-то, а все остальное — Алгол 68. Там же был и громадный вкладной лист с той самой синтаксической картой.

Еще я помню, что делал доклад про Алгол 68 в Доме Просвещения, где был даже Н. Амосов, и он задавал такие содержательные вопросы. Там даже не один доклад был, ведь мы занимались пропагандой Алгола довольно основательно. Уже после выхода “Кибернетики” с сообщением, осенью 70-го, я ездил по приглашению Рафика Мкртчана в Ереван и читал курс Алгола 68 в их университете. Под эгидой Мкртчана и Риммы Подловченко, которая там кафедрой заведовала. Потом, я еще здесь в НГУ тоже читал спецкурс.

Хорошо бы еще вспомнить ситуацию с Minority Report членов алголовской Рабочей Группы.

Рар: У меня об этом “Особом Мнении” что-то есть.

Берс: А у тебя, случайно, нет текста песни о Minority Report?

Рар: Случайно нет.

Берс: Вот, английский текст один человек по Интернету нашел, но он требует у нас русский текст, а я помню только фрагмент из припева

Подайте нам язык другой.
чтоб не было в нем обмана,
чтобы на нем могла писать
любая обезьяна...
Нет, нет, нет — это не наш язык.

Это было где-то зимой, и песню туда повез Маккарти, который до того у нас тут проживал.

Рар: Вот именно, это было в декабре 68-го во время празднования десятилетнего юбилея отдела программирования в ТБК. Там был и Маккарти, и там Ершов пел песню вроде бы на двух языках. А английский текст у меня есть.

Берс: Это надо сохранить. А русский? Может, у тебя и русский найдется, если хорошо покопаться... Интересно было бы найти... Мелодия эта или из Дилана, или из Джоан Баез ... Андрей Петрович увлекался тогда этой самой... — кантри.

Пар: А это наше с Цейтиным совместное письмо ван Вейнгаардену в июле 69 года, — вот тоже своеобразный перл: “Чтобы равенство в нашем равенстве сохранить...”

Берс: Ну да, а Minority Report... это 10 декабря 68-го, ладно, это я отдельно посмотрю просто-напросто... У нас к тому времени было готово предложение о национализации Алгола 68, ты его увез в Питер и там рассказывал, там оно одобрялось. Потом приезжал в Новосибирск М. Ламберт, старичок такой маленький, кругленький, симпатичненький... и мы с ним синтаксис поправок обсуждали. И всё это, естественно, повезли в Мюнхен.

Если уж говорить о Киеве, в частности о весеннем заезде 70-го года, то в связи с этими всеми обментами опечатками мы получили в Киеве исправления на исправления, то есть на опечатки. Я вспомнил анекдот про киевскую газетную опечатку в “Печерском листке” про коронацию, и мы его переводили на английский язык, чтобы порадовать ван Вейнгаардена. И это на самом деле удалось: в нашем английском варианте газета сначала сообщала, что на царскую главу была (вместо crown) возложена crow, а затем (в порядке исправления опечатки) — clown. В языке уже тогда было много семантики, перенесенной в синтаксис, хотя были и контекстные условия.

И мне кажется, что здесь стоит упомянуть выдающийся прорыв, который сделал Андрей Петрович, придумавший одновременную реализацию трех языков. Ведь, честно, А. П. сделал прекрасный шаг, он вытащил нас из языков. Каждый из нас воспитывался в каком-то алгоритмическом языке как программист, мы были в основном алголики, Курочкин был фортранец, и вдруг оказалось, что есть три больших языка, семантически достаточно похожих и достаточно разнообразных тем не менее. И А. П. решил, что мы их можем реализовать все вместе. И в результате, мы были вытащены из языка, и с этой точки зрения хорошо сравнить нас и А. Терехова, который реализовал Алгол 68, но в его стиле и застрял надолго...

И вот эта ситуация выхода в надъязыковое пространство представляется мне одним из выдающихся достижений А. П. Ершова. По крайней мере, в моей проектно-программной жизни выход этот сыграл очень большую роль. И с этой точки зрения я помню позицию, которую занимал Терехов. Например, безусловная статическая определенность языков и многое другое... Он за пределы идеологии Алгола 68 ну никак не мог вылезти, а мы спокойно вылезали. Вот хороший пример, реали-

зация матрицы двумерным массивом. Выяснилось, что в Фортране не было двумерных массивов, в Алголе были, в Алголе 68 двойной вырезкой можно было выбрать любой компактный минор, а в PL, если ты делал ограничения по одному измерению, то должен был ставить звездочку по другому измерению. То есть ты мог взять вертикальную полосу либо горизонтальную, а если хочешь взять минор, то надо взять из матрицы прямоугольный массив, назвать его новым объектом и потом взять опять прямоугольный массив. Но с другой стороны, разреженный минор, произвольный набор строк и столбцов (по Гантмахеру — знаешь ведь, что наряду с языками программирования есть такая хорошая книга, толстая — «Теория матриц») ни в одном языке программирования до сих пор взять нельзя. Причем есть ведь фундаментальный метод — метод прогонки, но ни в одном языке тип “матрица” не имеет диагонали в качестве подобъекта как сущности, то есть нельзя двигаться непосредственно и по диагонали, и по двум ближайшим диагоналям сразу, как этого требует прогонка.

И поэтому пришло понимание того, что такое подобъект, подтип. Потом Хоар объяснил всем, что есть структуры и в них выбираемые подобъекты — поля. А потом пошел Венский метод с селекторами на синтаксических и семантических деревьях... и предикаты. Хотя в “Пересмотренном сообщении” тоже появились предикаты в синтаксисе, и в результате всё резко упростилось и все контекстные условия выкинули. А я помню, как до предикатов мне пришлось написать две страницы синтаксических правил, чтобы породить конструкцию “*список понятий длины n* ”, где n задавалось как число в единичной системе (*iii iii i...*). Понятно, что конструкция эта порождалась из конструкций “*список понятий длины $n-1$* ”. Оказалось, что и при таком представлении чисел переносы между разрядами можно было спокойно таскать просто подбором метапонятий.

В 79 году вышел мой перевод “Пересмотренного сообщения”, и там удалось добиться того, чтобы можно было все синтаксические правила читать как предложения русского языка, причем достаточно естественные, благодаря находке и подсказке Цейтина, который предложил разделять слово “ПОЛЯ” как родительный падеж слова “ПОЛЕ” и слово “!ПОЛЯ” в качестве множественного числа. И, кроме того, мы выкинули слово “КВАЗИ” (в оригинале “ЕТУ”), заменив его знаком вопроса впереди метапонятия, например ?ДЛИННОЕ вместо LONGSETY. В результате все “portmanteaux” можно было элиминировать, и как я помню,

какой-то один я оставил, чтобы эпитафия сохранить. С эпитафиями я к тому времени хорошо наловчился работать, и я спокойно пользовался такой возможностью: слева стоит эпитафия ван Вейнгаардена — “The flowers that bloom in the spring, tra la, have nothing to do with the case”, а справа стоит “Каравай, каравай! кого хочешь выбирай!” к выбирающему предложению, но рубленным, чтобы ни с чем не спутать.

Примечание при корректуре: Оказывается, что все «хливые» метапонятия я таки в переводе “Revised Report” выкинул, а вместо эпитафии из Л. Кэррола “Well, ‘slithy’ means ‘lithe and slimy’. ...You see it’s like a portmanteau — there are two meanings packed up into one word” был поставлен эпитафия из Платоновского “Кратила”, который явно подчеркивал намерения переводчика: “Сократ: Однако, если ... имена должны быть выражением чего-либо, знаешь ли ты иной, лучший способ создать эти выражения, нежели сделать их возможно более тождественными тому, что они должны выразить?”. И это было одобрено и утверждено ВНТК!

Пар: За границей ради Алгола 68 я был еще в Дрездене, в 73 году. А Revised Report когда появился?

Берс: Примерно в это самое время.

Пар: Это могло быть именно там. На банкете ван Вейнгаарден, прозвонив тост, говорит, что купил сейчас в магазине книгу Гоголя в переводе на английский язык. Берет книгу и читает первую фразу: “Господа, я пригласил вас для того, чтобы сообщить вам пренебрежительное известие. К нам едет Reviser”. Да, это тогда было.

Берс: У ван Вейнгаардена естественно был замечательный стиль: сильный математик с чувством юмора. Умел подпустить крупицу соли.

Пар: Ну, это, наверное, характерно для всех.

Берс: Ну как сказать. Нет.

Пар: Костер, например.

Берс: Костер вообще под сильным влиянием Аада сформировался. Он, когда первый раз приехал, то вообще был еще мальчик, честно говоря... В общем, ван Вейнгаарден был очаровательным...

И сейчас можно проследить, что хотя сам язык не прижился, но от него осталось очень много идей. То, что типы можно описывать как угодно, какими угодно способами, потихоньку переросло в понятие класса. Это было не то подхвачено, не то, скорее всего, параллельно по-другому сделано в СИМУЛЕ, и потом потихоньку переключивалось аж до C++.

Вообще, само понятие вида аккуратно алгебраизировалось, причем множество значений перешло во множество операций, которые над ними можно выполнять. Далее имело место усиление выраженческого характера языка, и тут Алгол 68 был ведущий, я бы так сказал по своему опыту. Потому что возможность написать сложное выражение и по дороге что-то там же оставить, присваивая значение этого выражения, и теперь хорошо используется. Это тот самый вариант, когда один раз мы значение выражения использовали, а оно нам еще будет нужно, вот мы его и спрятали. Раньше для этого надо было выходить из хода рассуждений, причем не вычислений, а рассуждений, которые проводились.

С точки зрения сопоставления текста программы с ходом ее исполнения языка программирования ставили огромный барьер — буфер между тем, что написано, и тем, что делается. И повышение уровня языка программирования приводит к тому, что пишется одно, имеется в виду другое, а делается третье. Я на лекциях говорю, что поскольку программисты чаще всего не принцессы, то семантические горошины остаются незамеченными через “пятнадцать перин” трансляторов и операционных систем. Все это — тоже стиль Алгола 68.

Этот выраженческий характер был подхвачен в Smalltalk, в котором проявилось понятие “объект”. После появления “объектов” отслеживание системой изменений их состояния в динамике стало тем, что называется “программированием, управляемым событиями” — я вижу здесь подхват значения (как в выражениях), но при его изменении.

Конечно, очень интересно было сравнивать Алгол 68 с другими столь же мощными языками. У меня на полке стоит дипломная работа двух матлингвистов, сводный словарь перевода понятий для трех языков и попытка унификации их терминов. Вообще, с одной стороны было межъязыковое соперничество Алголиков и PL-щиков, но с другой стороны все это, как правило, касалось не фазы исполнения, а только языковых форм записи. Во всяком случае, с точки зрения тех людей, которые могли сравнить, проанализировать два или три языка вместе, — а это, конечно, разработчики Беты.

Но это, кажется, другая история...

Рар: Том третий, декомпозиция. “Руководитель Ершов, координатор первой фазы Берс, координатор второй фазы Поттосин, разработчик языковых форм Шварцман. 1972 год.” Это, кажется, эскизный проект.

Берс: С этой точки зрения мне представляется, что Алгол 68 сыграл

свою большую роль. Он был академическим языком, и так и не оказался производственным. Ленинградцы сработали очень сильно, сделали хороший транслятор, конечно, им они пользовались, но когда они над этим стали надстраивать технологию, у них это плоховато стыковалось со всем остальным.

И кстати, это выделение стандартного введения, которое составляет четверть описания языка. Если теперь взять какой-нибудь язык, BASIC For Application, который обслуживает Office, там языка-то мало, а всё остальное — это большая библиотека на каком-то языке, заголовки, которые называются функциями, конечно, назло всем математикам. Никакой однозначности отображения там в жизни не было. Вот в упорядоченности и ректификации свойств и понятий Алгол 68 сыграл большую роль.

И заодно, если мы уж разговариваем о мета-языковых вещах, то довольно интересным был подход венский, метод семантических деревьев, который описывал PL. Там достаточно аккуратно описывались связи значения с обозначениями. Это отслеживала Людмила Леонидовна Змиевская, она же отслеживала PL-стандарт. Алгол 68, кстати, тоже дошел до стандарта, до ГОСТа.

Пар: Но на этом и сдох.

Берс: Ну да. Хотя стоит отметить хорошую координацию работ по Алголу.

Пар: Да, была создана Временная научно-техническая комиссия, целая госкомитетовская эпопея. Она много сделала по стандартизации представлений, особенно для русской части.

Берс: Там работала бригада Маслова, они довольно сильно прошли по исходному тексту и значительно повлияли на само стандартное вступление в Revised Report. Потому что соответствующие процедуры Алгола 68 к тому времени проверились на существующих тогда уже трансляторах. Эта же самая комиссия помогала переводу Revised Report с одной стороны, а с другой стороны занималась прессингом, что я помню как переводчик. Однако некоторые наши новации пропали, например, “фонд” вместо “файл”.

Пар: Мы потерпели поражение в этой борьбе. Как-то пришлось согласиться с победителями.

Берс: Тем не менее, как было в русском языке всего два слова со слогом “айл”: кайло и хайло...

Пар: Но дело-то в том, что я в русском просто ненавидел слово

“файл”, не в английском, нет. Сейчас я к нему отношусь нормально. Заменяли его на “фонд”, но это не фонд все-таки. Сейчас у нас на персональном компьютере есть понятие “файл”, но это единица, некоторый единичный текст, и заменить его на “фонд” — это, конечно, совсем не то.

Берс: Да, но в конечном итоге не прошло и 25 лет, как все это привело нас к тому, что есть национальные версии систем программирования, операционных систем, — такие есть! Например, вот компакт-диск с MS Windows Russian Edition.

Рар: Ну да. Понятно. Теперь уже это понятно, значит, не зря...

Берс: Люди всегда будут думать-то на своем родном языке, но это пока не дошло до того, что появилась татарская национальная версия какого-то языка программирования. Кстати, у языков это как-то не прижилось, а вот что касается рабочих обстановок, операционных обстановок, то с ростом роли персональных компьютеров тут как-то ничего с одноязычием не вышло.

Рар: У глобалистов не вышло.

Берс: Да, у глобалистов. Языки программирования все продолжают быть англоязычными, просто, по-видимому, немцы и французы почему-то сдались. Хотя была интереснейшая вещь — французский вариант Алгола 68. Ведь там ещё была проблема в том, что при написании синтаксиса нужно было соблюдать определенный порядок слов.

Рар: У нас тоже ведь возникала такая проблема... Ну, что еще вспомнить...

Берс: Давай теперь поговорим о самом Алголе 68. На самом деле, все наши увлечения переводом не очень сильно и не очень интенсивно вели к разработке транслятора. Ведь транслятор с Алгола 68 сделали Цейтин и Терехов, а потом делала группа в Москве (Маслов). А мы только как-то отслеживали.

Что на меня в Алголе 68 произвело наибольшее впечатление? Во-первых, это был первый язык, в котором был очень богатый список средств описания типов, и можно очень четко и формально описать структуры с вариантами и скомбинировать с массивами и т.д., и т.п., и все, что по этому поводу проделывалось. Во-вторых, там был прекрасный механизм приведений, который позволял достаточно глубоко пользоваться описанными типами для того, чтоб перебрасывать по ним значения. И конечно, сама двухэтажная грамматика. И тут уже у меня возникло желание воспользоваться этим для того, чтобы делать хороший русский перевод.

Вообще в Алголе 68 была великолепно выдержана структура языка: словесные эквиваленты символов, закрывающие скобки для условных конструкций (например: *пока глушков в цк в киеве будет процветать кц все*), выбирающие предложения, параллельные фразы, хорошо проведенная борьба с побочными эффектами, хотя в языке они не запрещались.

Как любил говаривать ван Вейнгаарден: “не пишите такие программы”, и “если у вас есть язык, то на нем надо уметь выражаться”.

Да, еще одна замечательная вещь, которая там была, — это выдача значения почти любой конструкцией, что позволяло широко пользоваться суперпозицией при построении выражений. Важно осознать тот факт, что значение, которое выражение выдает, — однократно, неповторимо и уникально, и что оно может использоваться только двумя разными способами. Либо ты его подхватишь суперпозицией, либо должен спрятать как состояние объекта. В первом случае этому значению не надо давать обозначение, тогда его использование как бы от тебя скрыто. Потому что само это значение увидеть нельзя, и оно ничем не названо. С другой стороны, если ты его ничем не подхватишь, то оно исчезнет.

Сейчас в объектных языках всё очень хорошо: вы должны с помощью этого значения привести объект в определенное состояние. Приведя, вы можете из этого состояния вычитывать это значение столько раз, сколько вам надо, и получите, что оно обозначало. То есть вот эта композитная программа очень красиво позволяла писать синтаксические выражения очень связного характера. И поскольку языки программирования всегда тащили за собой английский жаргон, причем не язык, а именно жаргон... Тут еще одна вещь была, на мой взгляд, существенной — это работа над созданием русских национальных вариантов Алгола 68. Ершов и я, мы стремились к тому, чтобы все можно написать и без латинских букв, чтобы программы можно было бы практически читать вслух, — вещь недоступная для предыдущих языков. А синтаксис вообще удалось перевести как математический стих — все правила сохранили, и мнемонический смысл, и точную форму.

И в общем, язык получился довольно стройный. Но сложности ему добавляло, на мой взгляд, стандартное вступление. Это был еще более свернуто написанный программный текст с дополнительными надстройками макросного характера.

С. Б. Покровский
ВНУТРЕННИЙ ЯЗЫК В БЕТА

Проектные установки

Проект БЕТА был основан на предположении о конвергенции основных алгоритмических языков, как и методов трансляции и оптимизации. Подтверждением тому казалось значительное внешнее сходство тогдашних “хитов” — Алгола 68 и ПЛ/1, как и стоявшей несколько особняком Симулы-67.

Проект был задуман как сочетание научно-исследовательской и опытно-конструкторской работы, предпосылкой к этому должен был стать успешный опыт разработки Альфа-системы.

Входные языки

Первоначально основными входными языками считались Алгол 68 и ПЛ/1.

Сентиментальная привязанность к Алголу была обусловлена традицией, восходящей к Альфа-проекту и общей склонностью советской школы программирования к идейно чистым и элегантным построениям. Алгол 68 должен был стать стержнем всей конструкции, благодаря провозглашенным им принципам ортогонального, абстрактного (машинонезависимого) и алгоритмического (высокоуровневого) построения.

ПЛ/1 был данью прагматическим соображениям. Во многих отношениях он был прообразом Си — тоже своего рода гигантским языком ассемблера, долженствовавшим предоставить доступ ко всем возможностям базовой вычислительной системы (конкретно, ИБМ/360). С другой стороны, детище ИБМ, корпорации, которая занимала почти монопольное положение на рынке (хотя и в меньшей степени, чем теперь Майкрософт), ПЛ/1 имел и ряд монструозных черт, напоминающих нынешние Windows: эффектных, броских, кажущихся удобными — но по существу вредных (вроде определения атрибутов по умолчанию, которое, поощряя неряшливое написание программ, порой приводило к чудовищно нелепым результатам).

Следует также учитывать, что в течение ряда лет заказчиком работ по проекту БЕТА выступала организация, ответственная за реализацию ЕС ЭВМ (советского аналога серии ИБМ/360/370); на практике, однако, это не изменило второстепенного положения языка ПЛ/1 в проекте БЕТА.

С самого начала в списке входных языков фигурировала также Симула-67. Этот язык казался схожим со своими “большими братьями”, упомянутыми выше — как благодаря общему происхождению от Алгола 60, так и вследствие неотличения квазипараллелизма Симулы от средств управления многозадачностью языков Алгол 68 и ПЛ/1. На деле Симула, несмотря на свой консервативный синтаксис (она сохраняла обратную совместимость с Алголом 60), была языком совершенно другой природы, на два поколения впереди Алгола 68 и ПЛ/1 (языком следующего поколения стал Паскаль, но и в нем отсутствовали классы и префиксация Симулы, распространившиеся с языками объектно-ориентированного программирования).

Однако тогда Симула казалась интересным, но не очень важным языком.

Язык Паскаль появился на свет одновременно с началом работ над проектом БЕТА и вскоре был включен в список его входных языков как основной входной язык “пробной реализации” БЕТА-системы.

Не всё было честно в громадном успехе Паскаля, однако он бесспорно ознаменовал поворотный пункт от эры динозавров, вроде Алгола 68 и ПЛ/1, к эре мелких грызунов. До Паскаля каждый новый язык претендовал на расширение репертуара изобразительных средств, на всё большую вычислительную мощь — но уже Алгол 68 провозгласил принцип отбора, ортогонального построения. Паскаль выставляет своими приоритетами простоту для транслятора и ясность для программиста — требования взаимосвязанные и взаимодополняющие. На смену оптимизму и быстрым успехам 60-х пришли скучные пуританские добродетели 70-х гг. прошлого столетия.

В итоге от языкотворчества 60-х не осталось ничего. Парадоксально, но более живучими оказались языки, возникшие еще в 50-х: Фортран и Лисп — хотя и после многочисленных изменений, сходных с эволюцией естественных языков.

Внутренний язык

Центральным звеном проекта БЕТА был Внутренний язык, который должен был стать единым языком-посредником в БЕТА-системе, “наибольшим общим делителем” входных языков и “наименьшим общим кратным” выходных машин. Кроме этой своей роли промежуточного языка, позволяющего уменьшить число путей в схеме m -языковой n -машинной трансляции с $m \cdot n$ до $m+n$, внутренний язык должен был также явиться средой оптимизирующих преобразований, т.е. он еще должен был быть достаточно богат, чтобы на нем было возможно представить результаты оптимизации; например, экономии совпадающих подвыражений в операторе $a[i,j,k] := b[i,j,k] + c[i,j,k]$.

Разработка внутреннего языка шла в три этапа.

Вначале А. А. Берсу было поручено проработать “идеальный” внутренний язык, каким его желательно было бы видеть с точки зрения анализа программ, для выявления чистой сущности алгоритма обрабатываемой программы.

Параллельно М. И. Шварцман должен был разработать более надежный (более обыкновенный) запасной вариант. Однако первоначально “идеальный” вариант считался основным.

А. А. Берс построил интересную модель вычислений, управляемых потоком данных (информационными зависимостями), в значительной степени ориентируясь на задачу максимального раскрытия внутреннего параллелизма задачи (задача автоматического распараллеливания была одним из приоритетов проекта). Схема получилась интересная, и наверное она могла бы стать основой перспективного проекта, если бы она была ведущим элементом проекта, под который подбирался бы входной язык и конструировалась бы выходная машина. Однако в рамках БЕТА-проекта языки и машины были данностью, а внутренний язык не определяющим, но производным элементом, и было непонятно, как уложить, например, чрезвычайно жесткую, почти ассемблерную модель памяти ПЛ/1 в эту идеальную схему распределенной памяти.

Проект М. И. Шварцмана предполагал наличие изобразительных средств для таких конструкций низкого уровня, и в конце концов это определило смену приоритетов и переориентацию на более прагматический подход. Но инерция ли общего замысла проекта, или воспитание, или ориентация на научный результат (естественная ввиду диссертационных планов) — что-то резко изменило направленность работы М. И. Шварцмана, и с момента принятия его проекта за основу, он

быстро стал обрастать чертами идеально-научной разработки, хотя и не такой идейно чистой, как проект А. А. Берса.

Эта работа, однако, не была доведена до конца. М. И. Шварцман покинул проект, и дальнейшая ответственность за разработку внутреннего языка пала на С. Б. Покровского (автора этих строк).

Фактически работа была начата заново и на сей раз не “сверху вниз”, а “снизу вверх”. Было реализовано подмножество Фортрана, и оно дало некоторый каркас, на который понемногу наращивалась плоть соответственно с потребностями подключаемых входных или выходных языков или алгоритмов оптимизации. Язык был наконец зафиксирован в виде структур данных (в общем виде программа на внутреннем языке представляла собой граф, и линейно-текстовая форма для визуализации частей этого графа появилась лишь через некоторое время). Вместо идеальной модели вычислений, или универсального набора семантических сущностей для содержательной унификации, внутренний язык стал частично-интерпретированной схемой, что позволяло выполнять оптимизационные преобразования, но требовало конкретизации при подключении новых языков или машин. Ситуация довольно обычная в производственном программировании, но менее эффективная, чем было задумано.

Однако этот путь позволил реализовать часть задуманного, и маловероятно, что чистая полноинтерпретированная модель оказалась бы совместимой с заданным набором реальных входных и выходных языков.

Итоги

В итоге система БЕТА была реализована для языков Симула-67, Паскаль, Модула-2, Ада (подмножество) и выходных машин БЭСМ-6 и СМ-4. Был реализован скромный набор оптимизаций — несмотря на обширные замыслы, более скромный, чем в системе Альфа. В общем, сравнительно с Альфа-системой, проект БЕТА следует признать неудачным.

Отчасти тому были “субъективные” (т.е. личностные) причины; но также и сам замысел был слишком рискован и основан на неверных посылах.

Замысел был слишком рискован, так как одновременно предполагалось опробовать слишком много новшеств: многоязычие на входе, многомашинность на выходе; новые языки (например, при начале работ

существовали только неполные экспериментальные трансляторы с Алгола 68) и новые методы оптимизации для новых прикладных областей — и всё это в рамках научно-исследовательского института, полупроизводственная работа силами сотрудников, ориентированных на получение научных результатов.

Неверная установка состояла в экстраполяции успешного развития вычислительного дела конца 50-х — начала 60-х гг. Казалось, что после успешного освоения проблематики численных методов, для которой удалось создать удобные изобразительные средства и эффективные алгоритмы оптимизации, подобных же результатов можно будет добиться и в области символической обработки, системного программирования, искусственного интеллекта и т.д. Увы, эти области оказались более вязкими, а достижения в автоматизации программирования — гораздо более скромными, чем ожидалось в 60-е гг. Но этот просчет касается не только проекта БЕТА.

В целом, в подобной ситуации я бы рекомендовал яснее определять приоритеты и закладывать в проект меньше тонких мест. Другой вопрос, как при этом получить финансирование.

Л. В. Городняя

ЭКСПЕРИМЕНТАЛЬНЫЕ СИСТЕМЫ ПРОГРАММИРОВАНИЯ ЛИТТЛ, ЛИСП

Для многих учеников А. П. Ершова типичной темой дипломной работы была реализация какого-нибудь нового, модного, недоступного у нас или свежесочиняемого языка программирования. Некоторые разработки являлись чем-то вроде диалога с мировыми авторитетами в стиле интеллектуального вызова. Другие возникали как элементы технологии программирования в наших реальных условиях.

Лисп-система на БЭСМ-6

В конце 1968 года в кабинете А. П. Ершова Джон Маккарти прочел серию лекций, посвященных языку Лисп. Возможность практического применения такого языка даже на нашей лучшей ЭВМ БЭСМ-6 вызывала сомнения у специалистов.

Преодоление препятствий

Первая версия реализации Лисп-интерпретатора в Новосибирске была брошена после года работы из-за обнаруженного несовершенства исходных решений (хэш-таблица атомов оказалась непригодной к удалению объектов). Вторая версия экспериментальной реализации полного Lisp 1.5 в условиях получения пяти минут машинного времени в день, чего с трудом хватало на прогон одной колоды на автокоде, потребовала от Т. С. Янчук (вскоре мобилизованной в проект Альфа-6) изрядной изобретательности при выборе результативных проектных и технологических решений. Например, для представления интерпретатора она предложила специально сконструированный язык загрузчика, удачно сочетавший эффективность программирования в кодах с гибкостью символьной обработки. В результате за пять минут вполне удавалось пропустить более пяти колод по 1000 перфокарт. Протокол работы программы начинал печататься моментально, сразу вслед за завершением приема перфокарт. Это определило достаточную результативность отладки Лисп-интерпретатора.

Возможности Лисп-системы

Производственная версия Лисп-системы включала в себя русифицированную лексику, универсальную обработку свойств объектов и механизм перераспределения памяти с выгрузкой стека во внешнюю память. Алгоритмы и реализация были разработаны и отлажены Л. В. Городней. Л. В. Суковатицина (Черноброд) выполнила отладку Лисп-арифметики с целью поддержки будущих исследований в области верификации программ. Во время очередного визита в Новосибирск Джон Маккарти собственноручно написал тест для проверки функциональной полноты новосибирской реализации Лиспа. Система выдержала эту проверку.

Применение Лисп-системы

Лисп-система на БЭСМ-6 эксплуатировалась практически без изменений почти до 1986 года (до смены элементной базы). О ней имеется косвенное упоминание в книге “Мир Лиспа”, один из авторов которой бывал в Новосибирске и очень дотошно вникал во все, относящееся к Лиспу и его применению. Мюнхенский музей Лиспа в качестве экспоната располагает публикацией о новосибирском Лисп-проекте. Основное применение системы — исследования в области верификации программ, эксперименты по организации недоопределенных вычислений и конструированию семантических моделей.

Литтл-компилятор. Замысел реализации

Реализацию эффективной системы программирования для языка Set1 Дж. Шварц предполагал выполнить с помощью специально разработанного машинно-независимого машинно-ориентированного языка системного программирования Little, похожего на Fortran, но обрабатывающего произвольные коды и приспособленного к крупно блочной организации программ и данных. Предполагалось, что именно этот язык — подходящая основа для машинно-зависимого переноса программ на наши машины.

Идеи построения переносимого компилятора

Советская разработка системы программирования и компилятора для языка Little на БЭСМ-6 была выполнена Л. В. Городней в 1976 году. При реализации этой системы на макроассемблере было предложено

и экспериментально опробовано комплексное решение ряда технологических проблем обеспечения машинно-зависимого переноса программ без кросс-трансляции. Решение базировалось на идеях вертикального расчленения программ, функциональной декомпозиции на простые классы семантических систем, синтаксическое и семантическое управление конструированием компилятора, виртуального кода программы и оптимизирующей кодогенерации.

Вспомогательные инструменты

Попутно, чтобы читать программы с американских перфокарт, которые в отличие от наших (до появления ЕС ЭВМ) воспринимались узкой стороной, а не широкой, как на БЭСМ-6, был сделан перекодировщик (драйвер, как теперь говорят), отладка которого по сути дела была осуществлена с привлечением американских коллег в режиме почты того времени, т.е. две недели туда, две недели обратно в лучшем случае.

Заодно был разработан специальный формат листинга для компактной распечатки машинного кода с идентификацией точек входа, что было немаловажно, т.к. расход бумаги планировался на месяц вперед и лимит был достаточно жестким.

Для этого компилятора Л. А. Степанова реализовала условный макроассемблер, обеспечивающий интегрированное хранение машинно-зависимого текста программы для набора разных архитектур. Н. Н. Зыков, Н. П. Шиян и Н. А. Шупта реализовали три разных текстовых макрогенератора.

Сухой остаток

Опыт этой разработки дал материал для формализации реализации семантики языков программирования в виде интерпретируемых сетей, допускающих сетевое определение правила их функционирования. Кроме того, была выполнена систематизация требований ЕСПД к программной документации и представлена в форме, сходной с гипертекстом.

Д. Я. Левин

СЕТЛ: РЕАЛИЗАЦИЯ И ПРИМЕНЕНИЕ ТЕОРЕТИКО-МНОЖЕСТВЕННОГО ЯЗЫКА ПРОГРАММИРОВАНИЯ

Язык Сетл был предложен и разработан в начале 70-х годов профессором Нью-Йоркского университета Джекобом Шварцем (Jacob Schwartz). Это стало выдающимся событием: математик с мировым именем смело вошел в область программирования с радикальным практическим предложением. Оно состояло в том, чтобы, базируясь на множествах как основных типах данных, писать программы в терминах высокоуровневых непроцедурных алгоритмов, широко использующих логико-предикатные средства.

Вот два характерных примера записи на Сетле исполняемых алгоритмов.

Напечатать простые числа, меньшие N:

```
Print {n < N | not exist k, 1 < k < n | div (n, k)}
```

Выборка из базы данных:

```
Print { rec in DB | rec.age < 30 & rec.salary > good_salary}
```

В 1972 году Дж. Шварц по пути из Китая сделал короткую остановку в Новосибирске и рассказал о Сетле на семинаре в Вычислительном Центре. Организация этого выступления и последующего проекта Сетл в Новосибирске стали еще одним подтверждением выдающейся чуткости А. П. Ершова к фундаментальным тенденциям в программировании.

Формирование Сибирского проекта

Новосибирская Сетл-команда сформировалась естественным образом. Л. В. Городняя и Л. В. Черноброд к этому времени уже обладали огромным опытом работы с языком Лисп и его реализацией. Вкус к языку повышенного уровня, динамике данных, знание фундаментальных принципов и тонкостей этой области позволили им быстро оценить новый проект и энергично приступить к его реализации. Применение Лиспа для моделирования семантики Сетла имело свои преимущества и

недостатки. С одной стороны, операции с подвижными и иерархически множествами, включая необходимую для них сборку мусора (garbage collection), хорошо поддерживались базовыми механизмами Лиспа. С другой стороны, поскольку множества — это далеко не списки, то реализация ассоциативных выборок, построение теоретико-множественных объединений, пересечений, дополнений и др. в принципе не могли быть эффективно реализованы на Лиспе. В целом реализация на Лиспе сыграла ключевую роль прототипа, позволившего практически войти в проблематику Сетла.

В начале 70-х гг. А. А. Ляпунов неоднозначно относился к тенденциям программирования в целом, считая, что близорукая прагматическая составляющая оставляет слишком мало места для развития совершенно необходимой программированию подлинно системно-кибернетической и математической основы. Появление Сетла обрадовало и заинтересовало Ляпунова, известного своими фундаментальными результатами в теории множеств, до такой степени, что он откомандировал меня, своего бывшего дипломника, только что вернувшегося из армии сотрудника его лаборатории, “заниматься Сетлом к Ершову”.

Высокий профессионализм, творческий энтузиазм, прекрасная атмосфера интеллигентного коллектива, занятого в востребованной обществом сфере деятельности, интересная задача — все это, увиденное в Отделении Ершова, помогло мне быстро и практически включиться в проект Сетл. Опыт Лисп-реализации и полученные к тому времени результаты группы Дж. Шварца (BALM-SETL) показали, что продвижение к эффективной реализации нового языка лежит через построение специальной виртуальной машины, прицельно поддерживающей принципиальные элементы теоретико-множественного программирования. Создание такой машины, версии входного языка, компилятора и окружения — это стало моей задачей в 1973-75 гг.

Первые результаты

Сравнительно быстрому получению практических результатов способствовал выбор в качестве средства реализации Сетла системы Эпсилон—БЭСМ-6. Ее автор — А. Е. Хоперсков — отнесся к моей работе весьма неравнодушно, давая многочисленные тонкие советы и не жалея времени на разбор ошибок. Роль Эпсилона оказалась тем более важна, что почти в то же самое время в группе Шварца началось применение вполне аналогичного средства — языка LITTLE. Сегодня, в эпоху су-

губо интерактивной работы с компьютером, полезно вспомнить колоды с тысячами перфокарт, напряженное внимание к их прохождению через службу приема задач и машинный зал БЭСМ-6. Впрочем, это уже были далеко не те легендарные и по-настоящему героические времена 60-х годов: в 1973 году мне удалось в пакетном режиме совершить до 20-25 запусков в день...

Реализация Сетл—Эпсилон имела свои плюсы и минусы. С одной стороны, экспериментальная реализация так и не была доведена до профессионально сделанной системной программы, содержала много трюков, ошибок и др. С другой стороны, эта быстрая, прагматичная, иногда с изощренными алгоритмами реализация позволила почти сразу начать довольно широкое программирование на Сетле. Это привело к появлению эффектных примеров, практической пропаганде средства “very high level” (русский перевод всегда выглядел нескромно), стимулированию дальнейших работ и пр.

Это было прекрасное время безоглядной игры мускулами — возможностями очень быстро написать почти неалгоритмическую программу и наслаждаться ее работоспособностью. Было, например, проведено моделирование весьма модной в те годы fuzzy logic. Сделано это было на примере системы, которая по фразе типа “Нарисуй небольшой круг в правом нижнем углу экрана, а над ним — маленький квадрат” действительно печатала звездочками на АЦПУ довольно адекватную конфигурацию. Это была лишь игрушечная иллюстрация фундаментальной лингвистической модели “Смысл—Текст”, восходящей к известному советскому лингвисту И. А. Мельчуку, а также глубокого проекта “Рисунок—Информация—Текст”, выполнявшегося под руководством А. С. Нариньяни. Как раз в то время в Отделе Ершова находился некий докторант самого Заде: мне и сегодня приятно вспомнить, как был поражен иностранный гость, когда увидел, как вся основа fuzzy logic практически в том же синтаксисе записывается на Сетле и ... работает.

Примерно в 1975 году мне довелось провести несколько ознакомительных занятий по Сетлу для группы математических лингвистов НГУ. Умные и симпатичные студентки без затруднений писали короткие абстрактные алгоритмы в форме теоретико-множественных выражений, но не хотели верить в то, что “программы”, в которых отсутствовали изнурительные атрибуты “настоящего” языка программирования, могут непосредственно исполняться на ЭВМ. “Язык программирования — это ФОРТРАН”, — говорили опытные лингвистки.

В это время появились первые отечественные публикации о Сетле. Наряду со статьями о Сетл—Лисп и Сетл—Эпсилон большое значение имела работа В. Н. Касьянова, посвященная оптимизации. Действительно, “повышенный” уровень языка Сетл, новые типы данных, операции и языковые конструкции могли стать серьезным вызовом для развития и применения оптимизационной техники.

Поддержка и влияние корифеев

В 1974 году на Всесоюзной конференции в Бакуриани, при первом выходе в свет с Сетлом, я столкнулся с явным и глубоким интересом профессионалов к этому проекту. Этому способствовало и выступление В. М. Брябрина, который незадолго до этого провел несколько недель в Нью-Йоркском университете, и смог ярко рассказать об американском проекте. Вообще, отношение к Сетлу было всегда заинтересованное и благожелательное. Наряду с А. П. Ершовым — Берс, Брябрин, Котов, Лавров, Любимский, Тыгу, Поттосин, Фуксман и другие классики отечественного программирования сразу оценили стратегическую перспективу вклада, сделанного Джекобом Шварцем, и постоянно проявляли практический интерес к отечественному проекту.

Особое влияние на судьбу новосибирского проекта СЕТЛ оказал А. С. Нариньяни, когда эти работы стали вестись в рамках возглавляемой им Лаборатории Искусственного Интеллекта (ИИ). Главное состояло в том, что Александр Семенович предложил широко использовать Сетл для экспериментального программирования задач ИИ. Это был по-настоящему смелый и мудрый поступок: респектабельность в области ИИ предписывала пользоваться Прологом, а профессиональное программирование подразумевало безнадежную неэффективность языков высокого уровня. Позиция Нариньяни была свободна от консерватизма: Пролог намертво фиксировал как раз то, что должно было составлять суть экспериментов в ИИ, а “неэффективность” давно следовало трезво оценивать с позиций требований конкретных приложений. Вероятно, без определенного давления Нариньяни траектория Сибирского Сетла была бы гораздо более аккуратной и академичной. С сегодняшних позиций я без оговорок рад, что все произошло именно так, как произошло.

Советско-Американский Проект

Мне кажется, Джекоб Шварц оценивал неэффективность созданного им языка лишь в целом, равномерно и, как мне кажется, переоценил ее, чем надолго задержал крупные эксперименты по прототипированию и, тем более, построению приложений. Впрочем, свою миссию профессор Шварц выполнял и выполнил блестяще: она состояла, видимо, не в том, чтобы построить с помощью Сетла какие-то интересные прототипы, а в том, чтобы предложить в чистом виде новую фундаментальную парадигму программирования и в чистом же виде максимально полно показать ее проблемы.

К 1976 году в проекте SETL в Нью-Йорке участвовало не менее десяти прекрасно подготовленных сотрудников и аспирантов. Благодаря глобальному пониманию этого проекта Ершовым и поддержке Г. И. Марчука, было подписано межправительственное соглашение о нашем сотрудничестве в области реализации Сетл. Ярким и содержательным событием программы сотрудничества был Советско-Американский Сетл-семинар, состоявшийся ранней осенью 1976 года. Сначала была собственно формальная часть, проходившая в Москве, а затем большая часть американской делегации провела неделю в Новосибирске. На московском семинаре гости представили полную картину своих работ, сделав десять докладов; мы рассказали о своей реализации, а также об истории программирования в СССР; надолго запомнились ужины в ресторанах “Пекин” и “Славянский Базар”, а также выступление Э. Дейкстры, почти случайно заехавшего на наш семинар во время своего турне по СССР... В Новосибирске состоялись интенсивные сопоставления реализаций, оценка возможных приложений, обсуждение проблем.

Применение Сетла в Сибири

Первым заметным сибирским применением Сетла была реализация с его помощью прототипа системы общения с базой данных на естественном языке, выполненная в 1976 году. Прототип состоял из словаря, лингвистического процессора, переводящего естественно-языковые запросы к базе данных в некоторый стандартизованный язык СУБД, а также из собственно макета СУБД, включая и наборы демонстрационных данных. Лингвистический процессор был построен как компилятор, осуществляющий несколько проходов и основанный на продукционных правилах, которые принимали во внимание главным образом

несинтаксическую информацию. Макет СУБД хорошо подтверждал исходное предположение Шварца о естественности применения Сетла для работы с базами данных. У этой Сетл-программы, занимавшей не более двухсот строк и за 1–2 секунды отвечавшей на вопросы типа “Кто завлаб в лаборатории, где средняя зарплата не меньше 150 рублей?”, оказалась хорошая история. Во-первых, ее многочисленные варианты эффективно демонстрировались затем лет десять применительно к дюжине языков и нескольким предметным областям. Во-вторых, она послужила зерном, из которого выросли развитые обстановки для создания процессоров естественного языка. В-третьих, из нее выросла система СТЕНД — интегрированный инструментальный комплекс программирования.

В те годы мы в Лаборатории ИИ пропагандировали принцип оптимистического программирования: решать программистскую задачу на самом высоком языковом уровне, который допускает реальные требования данного приложения. Именно Сетл в то время дал фактическую возможность выбора между оптимистической и прочими позициями. СТЕНД — это была наивная, но благородная попытка построить “полную” инструментальную среду, в которой программист не должен был опускаться ниже уровня Сетла. Выше Сетла были производственные системы, СУБД, а самое главное — конструктор виртуальных процессоров. Наш аспирант, а впоследствии ведущий сотрудник, И. Е. Швецов прекрасно и чисто реализовал механизм универсального интерпретатора УНИК, который должен был работать в сочетании со специальными процессорами. В идеале программы виделись как суперпозиции троек процессоров: данных, вычислений и интерфейсов.

Кульминация и развязка

Примерно в 1977 году созрела идея сравнительно крупного международного эксперимента: в Национальном Центре Здоровья в Вашингтоне силами двух Сетл-групп построить высокоуровневую систему управления данными. В те годы такая идея требовала длительной бюрократической подготовки, которая при солидной поддержке Государственного Комитета по Науке и Технике (ГКНТ) была завершена к осени 1979 года. Нужно не по литературе знать тогдашние реалии, чтобы оценить и здесь проявившуюся раскованность А. С. Нариньяни, который сам предложил перенести нашу с ним двухмесячную поездку на последующую весну, ибо, по его мнению, зимняя одежда членов Советской делегации была на постыдном уровне самых низов американского народа.

Пожалуй, это была кульминация Сетл-проекта как в Новосибирске, так и в Нью-Йорке. В декабре 1979 случился “Афганистан”, после чего года три–четыре чиновник из ГКНТ, лелея мечту о действительно интересном международном проекте, безнадежно справлялся у нас, не могут ли наши американские коллеги нажать на Госдепартамент...

В дальнейшем наша работа с Сетлом была в основном связана с развитием вышеупомянутых прототипов, а также с серией переносов системы с БЭСМ-6 на другие машины (Burroughs 6700, ЕС ЭВМ, НОРД), совместными экспериментами с чехословацкими коллегами, некоторыми контактами с возникшим на короткое время проектом в Ростовском университете. Заметное оживление Сетл-деятельности наблюдалось в ходе легендарного проекта СТАРТ (1985-88).

Послесловие

А. П. Ершов никогда не менял своей точки зрения на Сетл как фундаментальное явление программирования. Важно, что он способствовал появлению черт Сетла в языковых средствах “детской информатики”, хорошо понимая их роль в освоении базовых понятий программирования.

Дж. Шварц, достаточно долго поддерживая интерес к своему детищу, затем все-таки перенес свой основной творческий интерес на другие области: сначала на робототехнику, а затем — на мультимедийные средства. (“Jack’s style has been to enter a new field, master quickly the existing research literature, add the stamp of his own forceful vision in a series of research contributions, and finally, leave behind an active research group that continues fruitful research for many years along the lines he has laid down.” — Martin Davis, Professor of Computer Science and Mathematics at NYU).

Хорошо понимаю, как мне повезло с Сетлом: интереснейшая задача, возможности карьеры, а самое главное — контакты с замечательными людьми.

Что я думаю про Сетл сегодня? Ценность теоретико-множественных и логико-предикатных элементов лексикона общения с ЭВМ не может устареть по определению. Другое дело, в какой контекст будут практически погружены эти элементы и какова будет архитектура их реализации.

Здесь необходимо вспомнить, что исходной идеей Шварца был язык абстрактных алгоритмов. С одной стороны, абстрактные алгоритмы

должны были реально исполняться и давать полноценную возможность прототипирования. С другой стороны, абстрактность алгоритмов подразумевала возможность их регулярной конкретизации. Уже в 1976 году в наших проектах появились библиотеки представлений множеств и методов работы с ними. В чем-то тот подход даже опережал время, но в нем был и принципиальный источник трудностей: отсутствие индустриальной базовой системы программирования и закрытость набора методов. Сегодня теоретико-множественная оболочка могла бы вполне продуктивно работать на основе C++ или Java...

А. В. Замулин

ИНФОРМАЦИОННЫЕ СИСТЕМЫ И БАЗЫ ДАННЫХ

В 1967 году А. П. Ершов поставил перед своим аспирантом А. В. Замулиным задачу исследования возможностей информационного поиска в режиме реального времени с обратной связью от пользователя к информационной системе. Этой инициативой было заложено новое направление работ отдела, продолжавшееся в дочерних подразделениях более 20 лет. В результате первого эксперимента в рамках системы разделения времени АИСТ-0 была создана прикладная программа ИПС-0, задачей которой являлся поиск библиографической информации в диалоговом режиме. В работе также принимала участие Г. С. Богданова. Эксперимент показал, что наличие обратной связи от пользователя к информационной системе в режиме диалога позволяет достичь одновременно и высокой полноты, и высокой точности поиска.

Параллельно с ИПС-0 в рамках проекта АИСТ была создана другая прикладная программа информационного характера — Информатор (автор В. Г. Котельникова), задачей которой было снабжение пользователей информацией о компонентах системы АИСТ-0 и способах работы с ними. Следует сказать, что эта программа являлась прообразом современных программ Help и была наиболее часто используемой программой системы АИСТ-0.

В связи с переводом проекта АИСТ в организованное в 1969 году КБСП дальнейшая работа по информационным системам продолжалась указанным коллективом в этой организации при тесном взаимодействии с отделом программирования. Следующей работой явилась информационная система КАДР, первоначально реализованная в КБ СП в 1972 году для нужд отдела кадров. Хотя по режимным соображениям система не была использована в этой организации, заложенные в ее основу элементарные возможности настройки позволили легко переориентировать ее на другое приложение и внедрить в областном управлении внутренних дел.

Опыт, полученный при разработке и внедрении системы КАДР, позволил сформулировать общие принципы построения информационно-поисковой системы общего назначения, пригодной для широкого круга

приложений и ориентированной на пользователя, не являющегося специалистом в программировании. В результате этого под научным руководством А. П. Ершова и непосредственным руководством А. В. Замулина в отпочковавшемся от КБ СП Новосибирском филиале Института точной механики и вычислительной техники АН СССР (НФ ИТМиВТ) была создана в 1974 году информационно-поисковая система ВЕГА, обладавшая развитыми средствами описания данных и манипулирования ими. Система использовалась длительное время в библиотеке НФ ИТМиВТ и других организациях страны. Фактически это была одна из первых автономных систем управления базами данных, положившая начало работам в СССР в области систем баз данных.

Опыт применения системы ВЕГА показал, что невозможно обеспечить пользователю приемлемый интерфейс, если не построить его с учетом потребностей и привычек данного пользователя. По этой причине А. В. Замулиным был предложен в 1975 году язык программирования баз данных БОЯЗ, предназначенный для построения конкретных информационных систем на основе универсальной системы программирования баз данных. В 1979 году в том же НФ ИТМиВТ была реализована система БОЯЗ-6 на ЭВМ БЭСМ-6. Она нашла широкое применение в стране для построения конкретных информационных систем и баз данных. В частности, в ВЦ СОАН была построена библиотечная информационная система, в ГПВЦ СОАН — информационная система контроля документов и система учета машинного времени, в Институте прикладной физики (Новосибирск) — подсистема учета кадров, в СНИИГГиМС (Новосибирск) — четыре базы данных по геолого-физической и управленческой тематике, в КБ им. Сухого (Москва) — подсистемы учета технической документации, учета средств измерения и учета кадров, в Институте металлургии АН СССР (Москва) — банк данных по свойствам фаз в двойных и тройных неорганических системах, на Красногорском механическом заводе (Московская обл.) — система учета движения материалов и распределения их по темам и подразделениям завода, база данных патентной информации и система обработки конструкторской документации. Следует отметить, что практически все коммерческие системы баз данных, созданные в 80-х и 90-х годах, стали предоставлять собственный язык программирования баз данных. В то же время язык БОЯЗ был первым в мире языком программирования баз данных, а система БОЯЗ-6 — одной из первых систем программирования баз данных.

В 1981 году, в связи с возвращением А. В. Замулина, в ВЦ СОАН в институте создается лаборатория систем программирования баз данных, в которой продолжают работы, начатые в НФ ИТМиВТ. В частности, используя систему БОЯЗ-6, Е. В. Пак провела большую работу по созданию библиотечной информационной системы в ВЦ СОАН, а аспирант С. С. Кобылов построил информационные системы для ГПВЦ СОАН. В 1986 году публикуется описание нового языка программирования баз данных АТЛАНТ, развивающего средства программирования БОЯЗа (в частности, воплощающего концепцию абстрактных типов данных), а в 1990 году заканчивается реализация системы на ЭВМ СМ-4. В работе принимали участие Г. А. Кучеров, В. А. Коситов, В. Д. Рыжков и Е. В. Пак. К сожалению, система не получила широкого внедрения в связи с быстрой сменой аппаратной базы и появления на рынке большого количества коммерческих продуктов для ЭВМ новых поколений.

Непосредственным продолжением работ по системам программирования баз данных явился проект расширения Алгола-68 средствами работы с базами данных. Работы проводились в ИСИ в 1990–1993 гг. коллективом в составе: А. В. Замулин (на начальном этапе), К. Д. Лерман (на начальном этапе), М. А. Бульонков и А. А. Бульонкова и в сотрудничестве с МПП “Терком” (Санкт-Петербург). В 1993 году созданная система была передана в Терком, где она поддерживается и развивается в настоящее время.

Л. В. Городняя

ОТКУДА БЕРУТСЯ ХОРОШИЕ ПРОГРАММИСТЫ

Пишу в связи с двадцатипятилетием Новосибирских летних школ юных программистов. Начиная издавека. В круг интересов академика А. П. Ершова естественным образом входили проблемы обучения программированию. Образовательный потенциал программирования А. П. Ершов отмечал еще в отчете о своей первой поездке в Англию. Знаменитый Альфа-проект, до сих пор встречающийся в ссылках, сопровождался впечатляющими краткосрочными курсами по программированию, заметно пополнившими ряды хороших программистов не только в Новосибирске. Уже в начале 60-х годов сотрудники отдела Ершова учили программированию как трудовой профессии школьников, многие из которых сохранили верность программированию до сих пор.

Школьная информатика

ШЮП, ЛШ, ЗШ, ШИ — этими аббревиатурами Андрей Петрович обозначал части механизма, продвигавшего Информатику и Компьютеры в наши школы:

- с 1975 года ЛШ или ЛШЮП (Летняя Школа Юных Программистов),
- с 1977 года ШЮП (Школа Юных Программистов),
- с 1978 года ЗШ (Заочная ШЮП),
- с 1982 года ШИ (Школьная Информатика).

Летом 1975 года Ю. А. Первин и Н. А. Садовская пригласили в Новосибирск из Харькова кружок юных кибернетиков Г. А. Звенигородского. Результативность харьковской методики вовлечения школьников в компьютерный мир настолько впечатлила А. П. Ершова, что в 1977 году Г. А. Звенигородский был приглашен в Новосибирск и сделал первый набор (более 70 школьников, возраст от 10 до 15 лет) в новосибирскую Школу Юных Программистов (ШЮП). Эту методику сейчас можно бы называть как объектно-ориентированное программирование учебных обстановок на понятном младшим школьникам языке Робик. В 1978 году на страницах журнала “Квант” Н. А. Юнерман энергично раз-

вернула Заочную Школу юных программистов (ЗШ), лучшие ученики которой приглашались в Новосибирск на Летние Школы (ЛШ) поработать на ЭВМ. Преподавали в этих школах преимущественно настоящие программисты-практики.

Летние Школы юных программистов

ЛШ — самое яркое и эффективное звено этого запущенного А. П. Ершовым механизма распространения и популяризации Информатики. ЛШ были предельно привлекательны для всех участников — школьников, педагогов, программистов, конструкторов ЭВМ и ученых.

На ЛШ съезжались школьники, чтобы впервые увидеть ЭВМ и вкусно поработать, педагоги, чтобы нащупать методику учебного применения ЭВМ, программисты, чтобы вместе со школьниками опробовать свои замыслы, конструкторы, чтобы испытать новую технику, ученые, чтобы популярно изложить свои идеи благодарной аудитории.

Ехали, чтобы за 10–20 дней узнать 5–10 языков и систем программирования, чтобы испытать себя в программистских олимпиадах, поработать в проектах рядом с мастерами программирования и по diskutieren с мировыми светилами и классиками программирования. Решения задач на олимпиаде представлялись любыми способами. Никому и в голову не приходило ограничивать используемые языки программирования. Делом чести компетентного жюри было обеспечить проверку решения на любом известном школьнику языке.

Ехали, чтобы вернуться знатоками программирования на ассемблере, Паскале, Форте, Алголе-68, Лиспе, Сетле, Модуле, Фортране, Поплане и т.д. и специалистами по наладке функционирования самого капризного в мире школьного оборудования: БЭСМ-6, HP, ЕС ЭВМ, СМ, Apple, АГАТ, Yamaha, BK0010, IBM PC, выставочные экземпляры зарубежной техники, предоставляемой ИПИ АН (А. В. Гиглавый). Машин остро не хватало — распределяли время круглосуточно.

Были случаи, что после ЛШ восьмиклассник в родном городе, где нет программистов, самостоятельно справлялся с установкой и запуском в эксплуатацию новых, никому не известных школьных компьютеров, превращая свою школу в очередной центр ШИ. Школьный учитель, прошедший ЛШ на правах ученика, получал импульс для самостоятельной организации аналогичной школы у себя дома.

Жизнь летней школы

Ехали, чтобы завести друзей, петь ночью у костра, ставить спектакли про Аду, Алгол и Фортран, “начинать” КВН, читать стихи, выпускать стенгазету и сочинять свои песни к конкурсу городов, слушать программистские байки от умных людей. А. А. Берс, Н. Н. Бровин, Д. Ш. Матрос, А. Н. Терехов, О. Ф. Титов, Л. Б. Штернберг и многие другие были доступны круглосуточно и в отличие от ЭВМ бесперебойно. Ежевечерние “Программистские сказки” А. А. Берса у костра впечатляли школьников не меньше, чем борения с ЭВМ. Иные лекции, например, по технологии программирования в исполнении А. Н. Терехова, завершались криком души юных дарований: “ХОЧУ!... ХОЧУ, ЧТОБЫ У МЕНЯ БЫЛИ ТАКИЕ УЧИТЕЛЯ!!”.

Сложнейшие приемы программотехники от кодирования до оптимизации программ, включая методы тестирования, разработки компиляторов и создания своих языков программирования и конструирования приложений, постигались школьниками ради продвижения любимых и интересных тем. Сообщение Л. Б. Штернберга, что он в любой программе найдет хотя бы одну неоптимальность, юные “ассы программирования” воспринимали как интеллектуальный вызов и рьяно доводили свои программы до совершенства.

Мир ЛШ отнюдь не замыкается на программировании. Вечерами ученые Академгородка, Москвы, Ленинграда и других городов много рассказывали о физике, биологии, генетике, лингвистике, астрономии, истории и другом. Рассказывали не только школьникам, но и педагогам и программистам.

Поощрение индивидуальности

Ученики могли показать себя не только успехами в учебе и программировании, но и докладом на конференции, и показом “домашних” проектов, и изданием газеты, и подготовкой концерта. Можно было поздравлять именинников на утренней линейке (Н. Н. Бровин ее проводил так, что это было весело), поболеть или победить на шахматном чемпионате, теннисном турнире, поиграть в баскетбол. На берегу Обского моря — катамараны, серфинг, водный парашют, прогулка на теплоходе.

Труднее всего жилось наградной бригаде (рук. Н. А. Юерман), экспертной комиссии (рук. О. Ф. Титов) и жюри (рук. Н. Н. Бровин). Экспертная комиссия добивается от юных программистов понимания, что

и зачем они делают на машине. Жюри выявляет, в чем именно школьник добился прогресса, чтобы при награждении каждый чувствовал, что его наградили заслуженно. Как-то не было принято бросаться словом “лучший”. Все, кто приехал на школу, имели возможность в чем-либо считать себя лучшими. Типичные формулировки награждения: “за достигнутые успехи ...”, “за интересную разработку ...”, “за красивое решение...”, “за самую сложную ошибку, найденную в ...”, “за удачную форму доклада ...”, “за любознательность на конференции и лекциях ...” или “за доставленное жюри удовольствие ...” (автор формулы Игорь Мадыаров из Иркутска). Поощрять старались индивидуально, отмечая все, что поможет юному программисту найти самостоятельный путь. Наградная бригада заботилась, чтобы призы соответствовали вкусам и способствовали профессиональному становлению растущих программистов.

Яркие события

Каждой ЛШ ее научный руководитель А. П. Ершов посвящал хотя бы один день, в 1987 году в работе ЛШ принял участие Дж. Маккарти, бывали и другие знаменитости. Все это создавало атмосферу энтузиазма, творческого накала, жажды знаний и доброжелательной инициативы. ЛШ начинали ждать сразу с сентября, как только приходили в себя.

Ученики и консультанты

Наиболее естественное формирование ЛШ получалось, когда приглашение учащихся определялось по результатам их учебы в ЗШ. Практиковался и конкурсный отбор, а также привлечение по рекомендации. Следует отметить, что рекомендации программистов-практиков намного превышают по уровню подтверждения в деле конкурсный отбор. Повидимому, критерии профессиональной пригодности к программированию еще не вполне осознаны, но на практике достаточно очевидны.

Если ЗШ занималась поиском наиболее способных и заинтересованных учеников для ЛШ (почти без возрастных ограничений, включая отдельных школьных педагогов), то Новосибирская ШЮП в течение учебного года формировала команду консультантов-подмастерьев, берущих на себя значительную нагрузку по организации и проведению ЛШ. Эта команда начинала работу с марта в форме заседаний оргко-

митета ЛШ. Обсуждались, проговаривались заранее, согласовывались разные детали механизма ЛШ — нечто вроде безмашинной отладки программ. Школьникам и студентам, включившимся в эту команду, многое доверяли, но и многое спрашивали. Ни возраст, ни неопытность, ни неумение не могли быть оправданием в случае провала поручения. Такая заранее сплоченная команда всегда была в курсе происходящего на ЛШ и обеспечивала ее здоровый микроклимат — дух ЛШ.

География участников ЛШ быстро расширилась: Харьков, Красноярск, Ленинград, Горький, Москва, Иркутск, Абакан, Братск, Томск, Петропавловск, Ульяновск, Махачкала, Владивосток, Фрунзе, Ташкент, Алма-Ата и другие города СССР, Болгария, Чехословакия, Венгрия, Германия, Польша, Голландия. По сути ЛШ превращалась в международный клуб любителей детского программирования.

Обучение программированию

Методика Г. А. Звенигородского раннего обучения программированию учащихся ШЮП, кроме языка Робик (идеи исполнителей которого отчасти воплощены в Роботландии Ю. А. Первина), активно использовала прорисовку синтаксических диаграмм, четкое выделение методически обусловленных учебных концентров и профилактику прирастания к первому средству. Разнообразие ЭВМ, языков и систем программирования, операционных систем, стилей и технологий представало перед учащимися в первый же год обучения. Для более старших школьников методика Г. А. Звенигородского в Новосибирске получила развитие в виде учебно-производственного языка Рапира — похож на бестиповый Паскаль с более универсальными структурами данных и управления — и проекта интегрированной учебной среды “Школьница”. Большую роль на этом уровне сыграла компьютерная графика — система Шпага для программирования прорисовки любимых картинок в понятных детям терминах.

Коллективные проекты

Важнейшей особенностью подхода Г. А. Звенигородского к обучению школьников следует признать целенаправленное формирование разновозрастных по квалификации и возрасту групп/мастерских, выполняющих коллективный проект по внешнему заказу. В группу включаются добровольно — кому интересно. При группе имеются консультан-

ты/подмастерья — знатоки используемых средств и методов из числа более опытных учеников. Группой руководит мастер — специалист, интересующийся задачами, близкими к внешнему заказу. Внешне это очень похоже на часто упоминаемый метод проектов: система вводных и базовых проектов ВКИ НГУ, проектные разработки по заказу учителей в своей школе, проекты по моделированию физических процессов (движение ракеты, “стрелялки” и т.п.). Принципиальная разница заключается в наличии внешнего заказа, критерии выполнения которого не подвластны руководителю и консультантам. Интересно, что прошедшие такую выучку программисты оказались в дальнейшем способны к бесконфликтному оптимальному распределению обязанностей в коллективных проектах.

Квалификация лучших учеников

Под руководством Г. А. Звенигородского группой студентов и старшекласников на первом советском ПК “АГАТ” была создана система “Школьница”, в которую вошли Робик и Рапира, использовавшиеся при разработке первых демонстрационных образцов программных средств обучения по школьной тематике. Эти образцы были продемонстрированы Комиссии очень высокого ранга, что и позволило ШИ стать новой школьной дисциплиной “Основы информатики и вычислительной техники”.

В 1985 году участники и выпускники ЛШ Л. С. Бараз, Н. Г. Глаголева, П. А. Земцов, Е. В. Налимов, Е. В. Боровиков, А. В. Грабарь, С. А. Терехов, Н. Ш. Погосян, М. А. Зайцев и др. под руководством Л. В. Городней приступили к профессиональной разработке школьного программного обеспечения на базе КУВТ Ямаха и Электроника УКНЦ. Кроме удобной системы программирования на языке Рапира с дружеским интерфейсом, окнами и достаточно быстрой машинной графикой, был разработан тренажер Микрорапира для начального знакомства с программированием, непревзойденный по устойчивости и удобству ряд текстовых редакторов Тог, система подготовки текстовых документов Дсрп, автоматизированные средства восстановления файловой системы Vfy, текстовый макрогенератор Grm, низкоуровневые средства ручной работы с файлами Fix и Nalim Commander для ПК Yamaha-MSX-2 — прекрасный аналог Norton Commander. В реализации на Электронике УКНЦ язык Рапира был обогащен возможностью работать с вычисляемыми именами и ассоциативными таблицами. Кроме того,

были разработаны эффективные сервисные средства системного администрирования и поддержки профессионального программирования на СМ-4. Многие из этих средств эксплуатируются в школах до сих пор.

Вывод

Таким образом Новосибирская система ЛШ+ЗШ+ШЮП+ШИ показала способность готовить программистские кадры мирового уровня. Механизм ее организации и функционирования вполне соответствовал критерию Дж. Вейнберга, автора основополагающего труда “Психология программирования”. На вопрос: “Откуда берутся хорошие программы и хорошие программисты?” — Дж. Вейнберг дает ответ: “Хорошее программирование, как хорошая дружба, крепнет от поощрения индивидуальности и признания достоинств.” Программные средства для новых компьютеров и очередные версии информационных систем многих известных фирм теперь выпускаются при участии выпускников ЛШ, ныне работающих в фирмах Intel, Microsoft, HP, IBM, Oracle и т.д.

Н. А. Черемных, Г. В. Курляндчик
БИБЛИОТЕКА И АРХИВ АКАДЕМИКА ЕРШОВА

История создания библиотеки

С самого начала своей научной деятельности академик А. П. Ершов очень много внимания уделял проблемам информационного обеспечения ученых.

В конце 50-х и в начале 60-х годов Ершов стал одним из немногих в то время “выездных” программистов. Он прекрасно знал английский язык, легко устанавливал контакты с зарубежными коллегами, а главное, понимал всю важность своевременно полученной информации. Принимая участие в международных конференциях, выезжая за рубеж в научные командировки, Ершов начал привозить литературу по программированию и смежным вопросам.

Бывая в ведущих научных центрах Европы и США, он договаривался об обмене информацией. Сам Андрей Петрович отправлял за границу много отечественных журналов, сборников, препринтов, а взамен на регулярной основе он получал научные отчеты из таких всемирно известных центров, как Массачусетский технологический институт, Стэнфордский университет, университет Северной Каролины (США), Национальная физическая лаборатория и Оксфордский университет (Великобритания), ИНРИА и Гренобльский университет (Франция), Технический университет Мюнхена (Германия), университет Торонто (Канада), университеты Швеции, Дании, Австралии и многие другие.

За тридцать лет активной научной деятельности А. П. Ершов установил личные, дружеские и профессиональные контакты практически со всеми классиками современного программирования, которые высылали ему свои книги, отчеты, препринты. Знаменитая книга Фр. Брукса “Мифический человек-месяц и другие очерки по технологии программирования” впервые появилась у Ершова еще в препринтном варианте, а уж затем автор прислал официальное издание.

И таких примеров можно привести множество.

Как член Американской ассоциации по вычислительной технике (ACM) и выдающийся член (Distinguished Fellow) Британского вычис-

лительного общества (BCS), он получал по подписке основные журналы, издаваемые этими организациями. Как член редколлегии или редактор, он получал такие международные журналы, как *Theoretical Computer Science*, *Acta Informatica*, *Information Processing Letters*, *Computers and Education*.

В знак уважения к А. П. Ершову как ученому и автору нескольких монографий, опубликованных в издательстве Шпрингер, это издательство бесплатно посылало ему свою известную серию *Lecture Notes in Computer Science*.

Таким образом, ко времени безвременной кончины А. П. Ершова в его личной библиотеке хранилось более 30 тысяч книг, журналов, трудов конференций, препринтов и отдельных оттисков статей практически на всех европейских языках.

Но что самое главное, это богатство не лежало мертвым грузом, доступ к нему имели без преувеличения все программисты Академгородка и многих других научных центров страны. В те годы, когда научно-техническая периодика, монографии и, тем более, отчеты из-за рубежа с трудом попадали в страну, библиотека А. П. Ершова была настоящим Клондайком — очень часто, однажды оказавшись в Академгородке и познакомившись с библиотекой, люди приезжали сюда снова и снова, чтобы в ней поработать.

Автоматизация библиотеки

В начале 70-х из очередной поездки в США Ершов привез более 400 различных препринтов, отчетов, отдельных оттисков статей. В этом массиве информации уже трудно было ориентироваться, возникла проблема его организации. Первую классификацию для библиотеки разработал сам Андрей Петрович, взяв за основу систему, используемую АСМ. В дальнейшем эта классификация дорабатывалась силами сотрудников Новосибирского филиала Института точной механики и вычислительной техники АН СССР под его руководством.

Одновременно встала задача автоматизации библиотеки. Первая информационно-поисковая система “ВЕГА” на базе библиотеки Ершова была создана силами сотрудников Новосибирского филиала Института точной механики и вычислительной техники АН СССР. В группу программистов под руководством А. В. Замулина входила В. Г. Котельникова. Эта система, реализованная на БЭСМ-6, обеспечивала поиск информации в массиве многих тысяч документов. Был разработан те-

заурус, позволявший вести поиск по ключевым словам. Все единицы хранения — т.е. книги, препринты, отдельные статьи из журналов, сборников, трудов конференций — были переведены в машинную форму, но параллельно сохранился и “ручной” каталог. Очень трудоемкая работа по составлению специальных карточек — описаний документов — была проделана группой сотрудников НФ ИТМ и ВТ, главным организатором и исполнителем всей этой работы была Г. В. Курляндчик.

ИПС “ВЕГА” позволила реализовать систему ИРИ (избирательное распространение информации по индивидуальным запросам специалистов), которая обслуживала абонентов из ВЦ, НФ ИТМ и ВТ, Института математики, Института автоматки в Академгородке, а также ученых из Киева, Минска, Иркутска, Красноярска, Москвы и Ленинграда, потому что в библиотеку А. П. Ершова поступали материалы, отсутствовавшие в книгохранилищах этих городов.

Мемориальная библиотека и архив А. П. Ершова

После смерти академика А. П. Ершова его наследники передали все собрание книг, журналов и документов в Институт систем информатики, который к тому времени выделился из Вычислительного центра. Так образовалась Мемориальная библиотека А. П. Ершова, вошедшая в Отдел научно-технической информации ИСИ.

Отдельный интерес представляет уникальный архив академика Ершова. Архив состоит из 500 с лишним толстых канцелярских папок, в которых хранятся документы, отражающие весь его жизненный путь, а вместе с ним — историю развития информатики в России. Каждая папка была собрана им лично либо по хронологическому признаку, либо по тематике.

В самых первых папках хранятся конспекты лекций выдающихся ученых, преподававших в МГУ во время его учебы там, черновики курсовых и дипломной работ, материалы к первым научным публикациям. Далее собраны документы, связанные с его переездом в Новосибирск и созданием Отдела программирования в Институте математики СО АН СССР.

Уникальны документы, связанные с проектом Альфа. Этот оптимизирующий транслятор с языка Альфа (отечественного варианта Алгола 60) для вычислительной машины М-20 стал первой крупной разработкой отдела программирования. В архиве сохранились журналы отладки Альфа-транслятора, в которых записи велись практически ежедневно.

К истинным раритетам можно отнести перфокарты программы для вычислительной машины “Стрела”.

Архив отражает огромную научно-организационную деятельность А. П. Ершова, который в разные годы жизни был председателем Научного Совета по комплексной проблеме “Кибернетика” АН СССР, руководил Комиссией по системному математическому обеспечению Координационного комитета по вычислительной технике АН СССР, Междугосударственной научно-технической комиссией по программному обеспечению ЭВМ при ГКНТ СССР и другими комитетами и комиссиями, во многом определявшими пути развития системного и теоретического программирования и всей вычислительной техники в нашей стране.

Все многочисленные зарубежные поездки А. П. Ершова отражены в отдельных папках. Поскольку он принимал самое активное участие в подготовке множества международных конференций и конгрессов, в архиве можно найти материалы его выездных дел, программы конференций и семинаров, переписку, касающуюся организации этих мероприятий, научные отчеты по командировкам, а кроме того — билеты, квитанции, записки для памяти — все эти материальные свидетельства давних событий, проходивших в самых разных точках земного шара.

Академик Ершов был редактором или членом редколлегии как русских журналов “Микропроцессорные средства и системы”, “Кибернетика”, “Программирование”, так и международных — *Acta Informatica*, *Information Processing Letters*, *Theoretical Computer Science*. В архиве собрана обширная переписка с редакциями и читателями этих изданий. Андрею Петровичу писали самые разные люди, многим он отвечал прямо или косвенно. Помимо редакционной переписки, обширный массив составляет научное эпистолярное наследие академика. Очевидно, нет ни одного известного в области информатики ученого, с которым бы А. П. Ершов не состоял в переписке.

Сохранились материалы публичных выступлений Ершова, во время которых он вел дискуссии с аудиторией. Записки слушателей свидетельствуют о живом интересе публики к новому научному направлению — созданию искусственного интеллекта.

Очень интересен процесс подготовки Международного симпозиума “Алгоритм в современной математике и ее приложениях”, отраженный в материалах архива. Симпозиум в Ургенче был задуман как научное паломничество на родину великого средневекового математика Аль-Хорезми, которому современная наука обязана самим появлением по-

нения “алгоритм”. Сохранилась переписка А. П. Ершова с участниками, среди которых было множество замечательных имен, варианты программы симпозиума, доклады, на нем представленные.

Подготовка и проведение крупнейших советских программистских мероприятий — а не было, наверное, за годы его активной научной деятельности ни одного, в котором А. П. Ершов так или иначе не участвовал бы, — также представлены многочисленными интересными документами. Материалы 2-й Всесоюзной конференции по программированию (ВКП-2), которая проходила в Новосибирске в 1970 году, говорят об огромном интересе к данной проблеме самых широких слоев советского общества. Архив содержит анкеты более 800 участников конференции и сотни заявок, многие из которых организаторы мероприятия не смогли удовлетворить.

Истории создания Конструкторского бюро системного программирования и Новосибирского филиала Института точной механики и вычислительной техники посвящено несколько папок, где можно найти переписку с соответствующими ведомствами, планы работ, биографические данные и пр.

Отдельный интерес представляют документы, касающиеся повседневной жизни научного коллектива, возглавлявшегося А. П. Ершовым. Это служебные записки, выписки из решений Ученого совета ВЦ, планы работ (по годам и на пятилетки). В папках сохранились списки сотрудников, направлявшихся “на картошку”, и объяснительные записки опоздавших на работу, письма в местный комитет с просьбой предоставить квартиру или место в детском саду для ребенка сотрудника отдела. В целом эти документы дают уникальную картину жизни советских ученых в 60–80-х годах.

Многочисленные папки посвящены собственно научной работе. В архиве сохранились рукописи практически всех статей и монографий Ершова. Не только историки науки, но и активно действующие ученые смогут найти в них “информацию к размышлению”, а, может быть, и почерпнут в старых, но не устаревших работах новые идеи или получат импульс к их появлению.

При финансовой поддержке корпорации Майкрософт силами сотрудников ИСИ сейчас осуществляется проект по переводу этого архива в электронный вид с тем, чтобы сделать его доступным через Интернет.

Руководство ИСИ приложило все силы к тому, чтобы сохранить библиотеку не только в знак памяти выдающегося ученого, но и как необхо-

димую составляющую всей научной деятельности. Удалось сохранить и закрепить многие контакты, продолжается обмен информацией со многими научными и университетскими центрами Европы и США. После организации в Академгородке локальной секции Американской ассоциации по вычислительной технике Мемориальная библиотека А. П. Ершова стала получать по подписке 45 журналов, издаваемых этой ассоциацией по всем направлениям вычислительного дела.

Три года помощь в подписке на издания АСМ и IEEE оказывал Международный институт технологии программирования при Университете Организации Объединенных Наций в Макао, который возглавлял тогда профессор Динес Бьорнер, коллега и друг Андрея Петровича.

В последнее время значительный вклад в развитие библиотеки внесла компания Microsoft Research Ltd. Благодаря ее финансовой поддержке, фонд библиотеки пополняется не только изданиями Microsoft, но и классическими работами по программированию, изданными за рубежом.

Вместо устаревшей системы, работавшей на БЭСМ-6, с помощью Новосибирского филиала Российского НИИ искусственного интеллекта и специалистов НФ ИТМиВТ в начале 90-ых годов создана БИБЛИОТЕКА — автоматизированная система хранения и поиска информации на персональных ЭВМ, которая на сегодняшний день обеспечивает доступ к массиву документов и позволяет автоматизировать многие библиотечные процессы. По-прежнему этой библиотекой пользуются сотрудники Объединенного Института информатики, программисты Института вычислительной математики и математической геофизики, Института математики, Института автоматики и электротметрии, Института органической химии СО РАН, а также специалисты из других центров СО РАН.

Е. И. Никольников

КОНСТРУКТОРСКОЕ БЮРО СИСТЕМНОГО ПРОГРАММИРОВАНИЯ — КБ СП (1968–1975)

Наука — для ПВО страны

Идея создания КБ СП в г. Новосибирске при Сибирском отделении АН СССР зародилась в 1966 году при посещении делегации ученых во главе с Председателем СО АН СССР академиком М. А. Лаврентьевым одного из крупнейших научно-исследовательских и испытательных центров страны в Казахстане. В составе делегации были видные математики и программисты (А. А. Ляпунов, М. И. Нечепуренко и др.). В ходе встреч с представителями НИИП и промышленных предприятий страны были сформулированы первоочередные задачи по совместному решению проблем в интересах ПВО страны, в частности, по использованию научного потенциала СО АН СССР для повышения эффективности создания программного обеспечения для комплексов ПСО и ПРО.

Решение о создании КБ

Совместные предложения Минрадиопрома СССР и СО АН СССР о создании в г.Новосибирске специального конструкторского бюро системного программирования были поддержаны Правительством СССР, и в мае 1968 года было принято Постановление Правительства о создании такого КБ на базе ВЦ СО АН СССР. В подготовке этого решения самое непосредственное участие принимали академики М. А. Лаврентьев, Г. И. Марчук, сотрудники ВЦ СО АН СССР А. П. Ершов, Г. П. Макаров, М. И. Нечепуренко, И. В. Поттосин, Г. И. Кожухин и другие сотрудники ВЦ, а также ряда Институты и служб ННЦ СО АН СССР. Головной организацией по решению возлагаемых на КБ СП научно-технических проблем было назначено НПО «Вымпел» Минрадиопрома СССР (г. Москва), и в частности, непосредственно на НИИ радиоприборостроения (директор Сидоров и главный конструктор — Г. В. Кисунко).

В феврале 1969 года в Академгородок прибыла делегация НИИРП, возглавляемая Кутеповым, для решения практических вопросов по созданию КБ СП (кадры, финансирование, строительство, сроки организации, взаимодействие с СО АН СССР и т.д.).

Первые руководители

20 мая 1969 года, через год после принятия первого Постановления Правительства СССР, в Новосибирск прибыл директор-организатор КБ СП представитель НИИ РП С. С. Московский, опытный специалист-разработчик систем управления ПВО, хорошо знающий проблемы, поставленные перед вновь создаваемой организацией. К тому времени приказом Министра радиопромышленности СССР уже были назначены по представлению руководства СО АН СССР главным инженером — первым заместителем директора КБ СП — Г. П. Макаров и заместителем директора по научной работе — А. П. Ершов (по совместительству).

Организационные вопросы

Вот эта “троица” и начала формирование коллектива КБ СП (структура, штатное расписание, персоналии и т.д.) и решение всех многочисленных и порой сложных технических, финансовых, хозяйственных и бытовых проблем новой организации. Поначалу дирекция ВЦ СО АН СССР приютила КБ СП в здании ВЦ, выделив несколько рабочих помещений и обслуживая сотрудников наравне со своими сотрудниками, а в сентябре 1969 года КБ СП получило отдельное здание на Детском проезде 7, раньше занимаемое Отделением кибернетики Института математики СО АН СССР.

К этому времени были решены основные организационные вопросы по структуре КБ СП, приему новых сотрудников, созданию управляющих структур (дирекция, Ученый совет, административные и технические службы, службы снабжения, охраны и хозяйственного обеспечения), а также определены основные научно-технические направления деятельности тематических подразделений КБ.

Основу научно-технического коллектива КБ составили сотрудники ВЦ СО АН СССР, выпускники мехмата НГУ 1967–1970 гг. и НЭТИ. Первыми начальниками тематических отделов стали М. И. Нечепуренко, Г. И. Кожухин, И. В. Поттосин, В. Л. Катков, технические подразделения возглавил Г. П. Макаров.

Учитывая важность проводимых в КБ СП работ, МО СССР организовало при КБ Представительство Заказчика, направив туда офицеров, непосредственно связанных с разработкой и испытанием перспективных образцов ПСО и ПРО. Первым начальником ВП был назначен И. М. Карпов.

В сентябре 1969 года директора-организатора КБ СП С. С. Московского сменил К. В. Тоболев, бывший начальник отдела программирования систем ПРО НИИП (Казахстан), а вслед за ним в феврале 1970 года начальниками тематических отделов были назначены И. А. Буряк и В. Н. Моисеенко, ранее занимавшие должности заместителей начальников управлений в том же НИИП и имеющие опыт по созданию и испытанию сложных комплексов ПВО.

Таким образом, к началу 1970 года, в основном, закончилось формирование коллектива сотрудников КБ СП и его обустройства в Академгородке, начались его рабочие будни по решению более сложных научных, технических, организационных, бытовых, строительных и психологических проблем, связанных со спецификой совместной деятельности с ЦНПО «Вымпел» (г. Москва) и СО АН СССР (ВЦ СО АН СССР), с одной стороны, и непосредственной работой на объектах ПВО (Казахстан), с другой стороны.

Противоречия и специфика деятельности

На протяжении последующих шести лет (1970–1975 гг.) деятельность КБ СП была связана с преодолением противоречий, заложенных как раз в этой специфике работы. Всегда возникал вопрос «Что главное и что делать?» Мнения разделились: одни считали, что главное — это создание средств автоматизации (инструментарий) для разработки прикладных программ сложных технических систем и моделирования процессов управления этими системами, а также оказание помощи в использовании этих средств тематиками (разработчиками) программного обеспечения, вплоть до участия в проведении испытаний проектируемых систем; другие же считали необходимым, кроме вышеозначенного, самим доводить программный продукт до окончательной реализации на объектах ПВО со сдачей программного обеспечения технических комплексов Заказчику.

Условия работы

Второе направление работ, безусловно более трудоемкое и сложное, связано с длительным взаимодействием сотрудников КБ СП с разработчиками образцов техники и программ в г. Москве, а также постоянным пребыванием этих же сотрудников на объектах (Казахстан) при отладке и испытании ПО, что в условиях неотлаженного транспорта и отсутствия минимально приемлемых бытовых условий работы сотрудников на объектах, представляло сложную проблему для руководства КБ СП.

Однако благодаря высокому профессионализму и патриотизму его сотрудников, задачи, поставленные Правительством страны, успешно решались. Среди сотрудников КБ было немало энтузиастов, месяцами находящихся в командировках, нередко по ночам отлаживающих программы на объектах в суровых условиях Казахской степи в жаркое лето и студеную и ветреную зиму.

В. А. Евстигнеев

**НАУЧНО-ОРГАНИЗАЦИОННАЯ ДЕЯТЕЛЬНОСТЬ
КОМИССИИ ПО СИСТЕМНОМУ МАТЕМАТИЧЕСКОМУ
ОБЕСПЕЧЕНИЮ ККВТ АН СССР**

1. ВВЕДЕНИЕ

К концу 70-х годов стало ясно, что необходимо координировать усилия разработчиков аппаратуры, программного обеспечения, прикладных программ общего назначения для устранения наметившегося отставания в этом важном научном направлении. С этой целью при АН СССР был организован Координационный комитет по вычислительной технике АН СССР под руководством тогдашнего Председателя СО АН СССР академика Г. И. Марчука. Особенностью этого Комитета были его состав и структура. В ККВТ кроме председателя и его заместителя чл.-корр. АН СССР Г. С. Поспелова входили еще 19 человек, девять из которых представляли Академию Наук СССР, остальные десять — министерства (Минрадиопром, Минэлектронпром, Минприбор, МинВУЗ и др.), а также Госплан и ГКНТ. Ранг всех членов ККВТ был достаточно высоким: от АН СССР академики и члены-корреспонденты, от министерств — заместители министров. Заметим, что представители министерств не пропускали (не игнорировали) пленумы ККВТ; более того, их активность возросла, когда Г. И. Марчук был назначен Председателем ГКНТ.

Каждый из девяти представителей Академии наук возглавлял Комиссию ККВТ, состоящую из наиболее известных ученых и специалистов из прикладных НИИ и учебных вузов в одной из областей вычислительных наук. Было выделено десять направлений, важнейшими из которых были архитектура ЭВМ (руководитель В. М. Глушков), системное математическое обеспечение (руководитель А. П. Ершов). Интенсивность работы комиссий была различная, одной из самых организованных была комиссия по СМО, вокруг которой сформировались

¹Работа выполнена при частичной финансовой поддержке РГНФ (грант 96-04-12030)

рабочие группы, занимающиеся конкретными проблемами.

Деятельность ККВТ проходила в виде пленумов, в работе которых принимали участие кроме членов ККВТ заместители председателей и ученые секретари комиссий и приглашенные специалисты. В работе пленумов принимал участие и Президент АН СССР академик А. П. Александров. Пленумы проходили достаточно бурно, нередко разворачивалась бурная дискуссия.

В задачи ККВТ входило создание сводных координационных планов по развитию вычислительной техники, разработке системного обеспечения и исследований в перспективных направлениях. Следует учесть, что создание ККВТ пришлось на время всеобщего перехода на серию ЕС ЭВМ, для которой не стоял вопрос о создании отечественного оригинального ПО со всеми вытекающими из этого проблемами. И хотя оценка решения прекращения большинства отечественных разработок ЭВМ и повсеместного распространения ЕС ЭВМ с копированием основного ПО была в большинстве случаев отрицательной, ставить так вопрос при анализе состояния дел и составлении координационных планов не рекомендовалось.

В данной работе предлагается описание деятельности Комиссии по СМО, которая под руководством А. П. Ершова стала центром, вокруг которого объединились все ведущие специалисты в области программирования и мнением которой, сформулированным в решениях Комиссии, дорожили. Решения Комиссии повлияли на принятие директивных постановлений на уровне министерств и правительства в целом. Работа выполняется в рамках создания информационной системы поддержки гуманитарных исследований, работа над которой ведется объединенным коллективом сотрудников Института систем информатики им. А. П. Ершова СО РАН и Новосибирского госуниверситета при финансовой поддержке Российского Гуманитарного Научного Фонда (РГНФ, грант N 96-04-12030).

2. СОСТАВ И СТРУКТУРА ККВТ АН СССР

В постановлении Президиума АН СССР об организации Координационного комитета по вычислительной технике указывается, что задачей ККВТ является координация фундаментальных исследований, проводимых в институтах Академии наук СССР и академий наук союзных республик, в следующих областях создания высокопроизводительной техники:

- архитектура вычислительных систем и комплексов,
- системное математическое обеспечение,
- организация банков данных и информационно-поисковых систем,
- сети ЭВМ и центры коллективного пользования,
- новая элементная база для средств вычислительной техники,
- требования к ЭВМ, математическому обеспечению и периферийному оборудованию.

2.1. Задачи ККВТ

Положением о ККВТ устанавливаются следующие задачи и функции Комитета:

1. Рассматривать и согласовывать планы работ, проводимых в АН СССР и академиях наук союзных республик в указанных выше областях.
2. Заслушивать отчеты институтов АН СССР по отдельным проблемам развития вычислительной техники; ежегодно готовить предложения о распределении целевым назначением финансовых и трудовых ресурсов между научными учреждениями АН СССР в пределах сумм, предусмотренных на эти цели в годовых планах АН СССР для обеспечения исследований по вычислительной технике.
3. Ежегодно готовить предложения о распределении ЭВМ, средств вычислительной техники и другого необходимого оборудования, в том числе поступившим по импорту, между научными учреждениями АН СССР и академиями союзных республик.
4. Готовить предложения:
 - о развитии работ в области вычислительной техники и ее использовании в научных учреждениях АН СССР и академиях наук союзных республик;
 - об организации новых учреждений;
 - о командировании за границу целевых групп специалистов для ознакомления с опытом работ и уровнем исследований в области вычислительной техники, а также давать рекомендации по персональному составу командируемых;
 - о подготовке специалистов для работы в области создания и использования высокопроизводительных ЭВМ, сетей ЭВМ и систем коллективного пользования.

2.2. Структура ККВТ

В состав ККВТ входят следующие комиссии и подразделения:

- комиссия по проблемам архитектур больших вычислительных систем,
- комиссия по системному математическому обеспечению,
- комиссия по вычислительным измерительно-информационным системам и комплексам,
- комиссия по вычислительным центрам коллективного пользования и сетям ЭВМ,
- комиссия по банкам данных и информационно-поисковым системам,
- комиссия по исследованиям проблем, требующих для своего решения ЭВМ большой производительности,
- комиссия по распределению и использованию вычислительной техники в АН СССР и координации работ по пакетам прикладных программ,
- комиссия по проблемам новой элементной базы для средств вычислительной техники,
- комиссия по применению ЭВМ в процессах управления,
- отдел вычислительной техники.

2.3. Состав ККВТ

Члены ККВТ от Академии наук:

Марчук Г. И. — академик, председатель;

Петров Б. Н. — академик, первый зам. председателя;

Глушков В. М. — академик, зам. председателя, комиссия 1;

Поспелов Г. С. — чл.-корр. АН СССР, зам. председателя, комиссия 9;

Дородницын А. А. — академик, комиссия 7;

Говорун Н. Н. — чл.-корр. АН СССР, комиссия 3;

Ершов А. П. — чл.-корр. АН СССР, комиссия 2;

Ржанов А. В. — чл.-корр. АН СССР, комиссия 8;

Русанов В. В. — чл.-корр. АН СССР, комиссия 6;

Стогний А. А. — чл.-корр. АН УССР, комиссия 5;

Якубайтис Э. А. — академик АН Латв.ССР, комиссия 4.

От министерств и ведомств в ККВТ входили: Ашастин (Госплан), Мясников (ГКНТ), Колесников (МЭП), Горшков (МРП), Кавалеров (Минприбор).

3. СОСТАВ И СТРУКТУРА КОМИССИИ ПО СМО

В ноябре 1978 г. в состав комиссии были включены следующие ученые:

Ершов А. П. — чл.-корр. АН СССР, ВЦ СО АН, председатель;
Любимский Э. З. — д.ф.-м.н., ИПМ АН, зам. председателя;
Евстигнеев В. А. — к.ф.-м.н., НФ ИТМ и ВТ АН, ученый секретарь;
Катков В. Л. — к.ф.-м.н., НФ ИТМ и ВТ АН;
Шура-Бура М. Р. — д.ф.-м.н., ИПМ АН;
Райков Л. Д. — к.т.н., НИЦЭВТ (МРП);
Лавров С. С. — чл.-корр. АН, ИТА АН;
Легичевский А. А. — д.ф.-м.н., ИК АН УССР;
Королев Л. Н. — чл.-корр. АН, МГУ (МинВУЗ);
Бабаян Б. А. — д.т.н., ИТМ и ВТ АН;
Вельбицкий И. А. — д.ф.-м.н., ИК АН УССР;
Курочкин В. М. — к.ф.-м.н., ВЦ АН;
Брябрин В. М. — к.ф.-м.н., ВЦ АН;
Шириков В. П. — д.ф.-м.н., ОИЯИ;
Тыугу Э. Х. — д.т.н., ИК АН ЭССР;
Барздинь Я. М. — д.ф.-м.н., ЛатГУ;
Поттосин И. В. — к.ф.-м.н., ВЦ СО АН;
Котов В. Е. — к.ф.-м.н., ВЦ СО АН.

Одна из особенностей комиссии — введение статуса наблюдателей. В отличие от просто приглашенных наблюдатель обладал правом совещательного голоса и являлся фактически кандидатом в члены комиссии. Первыми наблюдателями, которые были приглашены на заседание в Киеве в июне 1979 г. были М. Е. Неменман, Б. А. Головкин, М. Г. Цуладзе, Н. Н. Миренков, В. Ф. Хорошевский, П. Д. Румшас.

Предложение по составу комиссии на 1980 г. насчитывало уже 30 кандидатур. Первый список был дополнен следующими лицами:

Липаев В. В. — д.т.н., МНИИПА (МРП);
Лозинский Н. Н. — к.ф.-м.н., ЦНПО “Электронмаш”;
Макеев В. Г. — к.т.н., ЦНПО “Вымпел”;
Румшас П. Д. — ИМиК АН ЛитССР;
Неменман М. Е. — к.ф.-м.н., НИИЭВМ;
Чинин Г. Д. — НФ ИТМ и ВТ АН;
Корягин Д. А. — к.ф.-м.н., ИПМ АН;
Сафронов И. Д. — Минсредмаш;
Вантрусов Ю. И. — Минсредмаш;

Бреев В. П. — ГКНТ;

Амвросенко В. В. — НПО “Центрпрограммсистем”;

Ландау И. Я. — к.т.н., ИНЭУМ.

В состав Комиссии из этого списка не вошли И. Д. Сафронов, Ю. И. Вантросов, В. В. Амвросенко.

В качестве секретаря Комиссии была приглашена Д. А. Бухштаб (НИВЦ МГУ).

В 1982 г. Комиссия по СМО пополнилась рядом наблюдателей. От ВЦ АН СССР был включен к.т.н. В. П. Мазурик, от Кишиневского университета доцент Д. Н. Тодорой, от НИИ “Дельта” МЭП д.т.н. Д. Б. Подшивалов, В. В. Пивоваров (к.т.н., в/ч 03425), Е. Ф. Гусев (секция прикладных проблем АН СССР). Был переведен из наблюдателей в члены комиссии М. Г. Цуладзе. В итоге к концу 1982 г. Комиссия по СМО включала 28 членов и 8 наблюдателей. Ряд членов Комиссии и наблюдателей практически так и не участвовали в работе Комиссии (В. П. Бреев, Н. Я. Матюхин, Г. Л. Столяров, Б. А. Головкин). В начале 1984 г. в состав наблюдателей был включен В. П. Котляров (ЛПИ). В 1985 г. в состав Комиссии был включен чл.-корр. АН СССР В. П. Иванников (НИИ “Дельта”), В. И. Саликовский (ГКНТ СССР) и в качестве наблюдателя А. Л. Александров (НИВЦ МГУ).

4. ХРОНИКА ДЕЯТЕЛЬНОСТИ КОМИССИИ ПО СМО

Отчеты о деятельности Комиссии по СМО были опубликованы в 1984 г. [1] и в 1986 г. [2]. Краткие отчеты о деятельности Комиссии публиковались в Бюллетене ККВТ АН СССР.

Программа заседаний Комиссии обычно состояла из двух частей: знакомство с состоянием дел в институте, группе институтов, регионе и обсуждение состояния дел в некотором научном направлении информатики. По итогам обсуждения принимались соответствующие заключения. Все заседания подробно протоколировались, утвержденные протоколы направлялись всем членам и наблюдателям Комиссии, а также в отдел вычислительной техники Президиума АН СССР для последующей публикации в сокращенном виде в Бюллетене ККВТ АН СССР.

4.1. Первое заседание (6 февраля 1979 г.)

Первое заседание проходило в Москве (МГУ). Повестка дня включала пять вопросов.

1. Задачи Комиссии и порядок ее работы.
Докладчик — А. П. Ершов.
2. Рассмотрение программы исследований по важнейшим фундаментальным исследованиям на период 1980–1990 гг. в части системного математического обеспечения.
Докладчик — А. П. Ершов.
3. Рассмотрение поправок к Договору о сотрудничестве между АН СССР и Минрадиопромом и вопросов координации с организациями Минрадиопрома СССР.
Докладчик — В. Л. Катков.
4. О подготовке доклада “Математическое обеспечение средств вычислительной техники” на мартовском пленуме Координационного комитета АН СССР.
Докладчик — А. П. Ершов.
5. Постановочные вопросы.

4.2. Второе заседание (4 июня 1979 г.)

Второе заседание Комиссии проходило в Киеве в Институте кибернетики АН УССР. Повестка дня включала четыре вопроса.

1. О состоянии исследований в области системного математического обеспечения в АН СССР и АН союзных республик.
Докладчик — Э. З. Любимский; содокладчики — А. А. Летичевский, В. М. Курочкин, И. В. Поттосин, Д. А. Корягин, Б. А. Бабаян, В. М. Брябрин, В. Ф. Хорошевский.
2. О проекте плана работы Комиссии по системному математическому обеспечению на 1980–1982 гг.
Докладчик — А. П. Ершов.
3. Информация о координации работ АН в области системного программирования с Минприбором, МЭП и Минвузом.
Докладчики — И. Я. Ландау, Л. Н. Королев, А. П. Ершов.
4. Разное.

4.3. Третье заседание (4–7 декабря 1979 г.)

Третье заседание Комиссии проходило в Новосибирске в ВЦ СОАН СССР. Повестка дня включала четыре вопроса.

1. О состоянии работ в области технологии программирования.
Докладчики — Г. Д. Чинин, И. В. Вельбицкий.

2. Рассмотрение предложений к проекту постановления Правительства СССР в части системного математического обеспечения.
Докладчик — А. П. Ершов.
3. О программе совместных работ АН СССР и организаций Минприбора и Минвуза.
Докладчики — А. П. Ершов и представители соответствующих ведомств.
4. Разное.

4.4. Четвертое заседание (3–4 июня 1980 г.)

Четвертое заседание Комиссии проходило в Киеве в Институте кибернетики АН УССР. Повестка дня включала четыре вопроса.

1. Отчет Института кибернетики АН УССР по работам в области системного программирования.
Докладчик — зам. директора чл.-корр. АН УССР А. А. Стогний.
2. Языки и системы программирования.
Докладчики — И. В. Поттосин, М. Р. Шура-Бура, В. В. Липаев, Л. А. Серебровский.
3. О взаимодействии с Комиссией по пакетам прикладных программ.
4. Разное.

Перед обсуждением первого вопроса с вступительным словом выступил А. П. Ершов. Он уточнил смысл предстоящего отчета Института кибернетики, указав, что главное — подробно ознакомиться с работой института и сопоставить результаты, достигнутые институтом, с требованиями народного хозяйства и с мировым уровнем.

По итогам обсуждения было принято *Заключение*.

Перед обсуждением второго вопроса А. П. Ершов сказал: “Данный вопрос — основной для нашей Комиссии, по данной тематике наработан большой опыт, поэтому результаты работы Комиссии будут иметь большое значение. В дискуссии хотелось бы услышать мнения по ряду вопросов.

- 1) Есть ли единый взгляд на ситуацию с языками программирования?
- 2) Как делать трансляторы?
- 3) Как ликвидировать дефицит литературы?
- 4) Как хорошо учить системному программированию?”

4.5. Пятое заседание (17–18 февраля 1981 г.)

Пятое заседание Комиссии проходило в Москве в Институте прикладной математики им. М. В. Келдыша АН СССР. Повестка дня включала следующие вопросы:

1. Отчет Института прикладной математики им. М. В. Келдыша АН СССР в области системного программирования.

Докладчик — чл.-корр. АН СССР В. В. Русанов.

2. Системное математическое обеспечение пакетов прикладных программ.

Докладчики — М. М. Бежанова, Д. А. Корягин, Э. Х. Тыгу.

3. О плане дальнейшей работы Комиссии.

По первому вопросу содокладчиками выступили заведующие отделами М. Р. Шура-Бура, В. С. Штаркман, И. Б. Задыхайло, В. А. Крюков, Э. З. Любимский.

По второму вопросу были заслушаны доклады:

М. М. Бежанова “О состоянии работ в области ППП за рубежом”,

Д. А. Корягин “О состоянии работ в области системного обеспечения пакетов прикладных программ в СССР”,

Э. Х. Тыгу “Способы управления пакетами прикладных программ”.

4.6. Шестое заседание (21–22 мая 1981 г.)

Шестое заседание Комиссии проходило в Новосибирске в Вычислительном центре СО АН СССР. Повестка дня включала три вопроса.

1. О работах Вычислительного центра СО АН СССР в области системного математического обеспечения.

Докладчик — зам. директора В. Е. Котов.

2. Операционные системы для ЭВМ и МВК.

Докладчики — Б. А. Бабаян, Л. Б. Эфрос, И. Я. Ландау, Л. Д. Райков.

3. Разное.

По первому вопросу содокладчиком выступил директор НФ ИТМ и ВТ АН СССР Г. Д. Чинин с докладом “Об истории создания НФ ИТМ и ВТ и о совместных с ВЦ СО АН СССР работах”.

По второму вопросу были заслушаны доклады:

Л. Б. Эфрос “Актуальные проблемы создания операционных систем для современных ЭВМ и вычислительных комплексов”,

- Б. А. Бабаян* “История развития ОС в СССР и современное состояние вопроса”,
И. Я. Ландау “Исторический обзор развития работ по операционным системам в Минприборе”,
Л. Д. Райков “О состоянии дел в области операционных систем ЕС ЭВМ”.

4.7. Седьмое заседание (15–16 декабря 1981 г.)

На декабрьском заседании 1981 г., проходившем в Москве в Вычислительном центре АН СССР, был заслушан отчет ВЦ АН СССР по работам в области системного математического обеспечения. Научная программа включала в себя обсуждение состояния дел в теоретическом программировании. Этот отчет завершил серию заседаний Комиссии, на которых были заслушаны доклады о деятельности академических институтов, внесших наибольший вклад в становление программирования в СССР — Института Кибернетики АН УССР, Института прикладной математики АН СССР, Вычислительного центра СО АН СССР, Вычислительного центра АН СССР.

По п.1 повестки дня с докладами о деятельности Вычислительного центра АН СССР выступили зам. директора д.ф.-м.н. Ю. Г. Евтушенко и ведущие специалисты Института В. М. Курочкин, В. М. Брябрин, М. А. Копытов, И. Е. Педанов, В. П. Мазурик.

По научной программе заседания (п. 2 повестки дня) были заслушаны доклады:

- В. А. Непомнящий, И. М. Дехтярь* “Математическая теории программирования. Обзор зарубежных работ”,
А. А. Летичевский “Математическая теория программирования. Состояние дел в СССР”.

Комиссия, отметив успехи советских ученых в данном направлении, сочла недостаточным влияние достижений теории на практику конструирования программного обеспечения, в том числе на промышленные разработки. Недостаточно также, с точки зрения Комиссии, развиваются такие важные для практики направления как теория сложности алгоритмов, теория рекурсивного программирования, теория абстрактных типов данных, логические методы программирования. В своем решении Комиссия рекомендовала расширить фронт фундаментальных исследований по математической теории программирования в академических институтах, сосредоточив основные усилия на следующих направлениях:

- 1) разработка и теоретическое обоснование практических методов и алгоритмов решения задач анализа и синтеза, а также оптимизации алгоритмов в системах математического обеспечения современных и перспективных ЭВМ;
- 2) построение и исследование новых моделей вычислений и вычислительных систем с учетом перспектив развития архитектуры и технологии проектирования ЭВМ и систем математического обеспечения;
- 3) разработка теоретических проблем проектирования математического обеспечения для решения отдельных классов задач (вычислительной математики, обработки данных, искусственного интеллекта и т.п.).

4.8. Восьмое заседание (10–11 марта 1982 г.)

На мартовском заседании 1982 г., проходившем в Таллине в Институте кибернетики АН ЭССР, были заслушаны отчеты прибалтийских Академий наук по работам в области системного программирования; научная сессия была посвящена вопросу о методологии программирования.

С отчетами выступили чл.-корр. АН ЛитССР Л. А. Телькснис, д.ф.м.н. Я. М. Барздинь и чл.-корр. АН ЭССР Э. Х. Тыугу. Комиссия отметила достижения ученых прибалтийских республик в исследованиях по теоретическим вопросам программирования, пакетам прикладных программ (Литва), в исследованиях по системному математическому обеспечению для вычислительных сетей, по индуктивному синтезу программ и автоматизации тестирования (Латвия), в исследованиях по автоматическому синтезу программ, сложности алгоритмов, по языкам программирования и трансляторам (Эстония).

По п. 2 повестки дня в рамках научной программы были заслушаны доклады:

- чл.-корр. АН СССР С. С. Лавров* “Методология программирования”,
Б. Г. Чеблаков “Влияние идей Парнаса на методологию программирования”,
И. Р. Агамирзяна А. С. Иванов, А. В. Проскурин “Механизм абстракции в расширяющихся системах”,
Н. Н. Непейвода “Методология программирования”.

4.9. Девятое заседание (4–8 октября 1982 г.)

На октябрьском заседании 1982 г., проходившем в пос. Пасанаури близ Тбилиси, были заслушаны отчеты Академий наук Азербайджанской, Армянской и Грузинской республик. С отчетами выступили Т. М. Алиев (АН АзССР), С. Т. Хачатрян (АН АрмССР), М. Г. Цуладзе (АН ГССР). Комиссия отметила в целом интенсивный характер ведущихся исследований по системному программированию, наличие ряда интересных разработок. К таким результатам относились в первую очередь создание в ВЦ АН АрмССР) системного и математического обеспечения автоматизированной системы коллективного пользования для научных учреждений АН АрмССР, работы по созданию трансляторов, информационно-поисковых систем, проблемно-ориентированных языков, пакетов прикладных программ (ВЦ АН ГССР).

Научная программа заседания включала следующие доклады, посвященные проблемам создания программного обеспечения задач искусственного интеллекта:

- В. М. Брябрин* “О проблематике задач искусственного интеллекта и о состоянии исследований по искусственному интеллекту в СССР”,
В. Н. Пильщикова “Языки и системы программирования для задач искусственного интеллекта и состоянии исследований за рубежом”,
Ю. Д. Петросян, Л. Л. Цинман “О проблемах машинного перевода и об экспериментальной системе машинного перевода ЭТАП-1”.

Дискуссия показала, что исследования в области искусственного интеллекта ведутся с большой интенсивностью во всем мире, особенно в Японии и США. В Советском Союзе создан ряд систем искусственного интеллекта, относящихся к лучшим на мировом уровне, однако при их разработке применялись языки программирования, не ориентированные на этот класс задач, а использование средств высокого уровня ограничено малым числом использующих их коллективов и недостаточной производительностью машин.

Комиссия пришла к выводу, что исследования в области искусственного интеллекта составляют одно из важнейших направлений разработки человеко-машинных систем; необходимо дальнейшее развитие фундаментальных и прикладных исследований в этой области, в том числе фундаментальных исследований по прикладной (вычислительной) лингвистике и исследований по созданию прикладных систем, основывающихся на методах искусственного интеллекта (интеллектуальные системы общения с базами данных, интеллектуальные пакеты приклад-

ных программ, экспертные системы для разных предметных областей, интеллектуальные роботы, лингвистические процессоры). Необходимо также создание систем общения на основе современных дисплеев и специализированных процессоров.

4.10. Десятое заседание (18–22 апреля 1983 г.)

Десятое заседание Комиссии проходило в апреле 1983 г. в г. Дубне. Заседание было совместным с Комиссией по вычислительным измерительно-информационным системам (председатель — Н. Н. Говорун). В повестке было три вопроса:

1. О работах в области системного программирования, проводимых в физических центрах СССР.
2. О программном обеспечении микропроцессорной вычислительной техники.
3. О программном обеспечении роботов.

По первому вопросу с докладами выступили чл.-корр. АН СССР Н. Н. Говорун (ОИЯИ), М. А. Ташимов (ИФВЭ АН Каз.ССР), А. Н. Румянцев (ИАЭ им. И. В. Курчатова, Ю. Ф. Рябов (ЛИЯФ АН СССР), В. Ф. Жильченков (ИФВЭ), В. С. Криворученко (ЦАГИ), В. Л. Крейсберг (ВНИИ “Теофизика”).

По второму вопросу выступили:

- Е. Е. Дудников (МНИИПУ)* “Зарубежные домашние и учебные компьютеры”,
В. М. Брябрин (ВЦ АН СССР) “Профессиональные персональные компьютеры”,
Г. Р. Громов “Индустрия ЭВМ и персональные компьютеры”,
И. С. Лосев “Операционные системы для персональных компьютеров”,
А. В. Гизлавый, И. Я. Ландау “Микро-ЭВМ линии СМ”,
А. Ф. Иоффе, А. Ю. Кривцов “Вычислительный комплекс «Агат»”,
А. Б. Либеров “Автоматизация программирования для микро-ЭВМ”,
В. В. Липаев, Ф. А. Каганов “Адаптируемые кросс-системы для проектирования программ микро-ЭВМ на технологической базе универсальных и персональных ЭВМ”,
И. В. Вельбицкий “Сообщение о работе целевой подгруппы микропроцессорной техники Рабочей группы ГКНТ по технологии программирования”.

После обсуждения Комиссия приняла решение, в котором подчеркивалось, что необходимо:

- 1) развернуть работы по созданию отечественных ПК всех типов;
- 2) создать системы автоматизации проектирования и программирования микро-ЭВМ;
- 3) форсировать работы по созданию системного и прикладного ПО профессиональных ЭВМ, ориентированного на пользователей, не являющегося профессиональным программистом;
- 4) уделять первоочередное внимание организации сети сервиса ПЭВМ;
- 5) рекомендовать МЭП согласовывать с Комиссией по СМО разрабатываемые перспективные БИС, предназначенные для использования в микро-ЭВМ;
- 6) интенсифицировать создание систем автоматизации научного эксперимента на базе микро-ЭВМ;
- 7) форсировать работы по созданию локальных сетей микро-ЭВМ;
- 8) наладить крупносерийное производство микропериферийных устройств (цветных мониторов, накопителей на гибких магнитных дисках, знаковосинтезирующих принтеров, жестких минидисков типа “Винчестер”, устройств графического управления типа “Джойстик”, “Маус” и др.);
- 9) популяризовать и распространять информацию о возможностях микро-ЭВМ и ПЭВМ и областях их применения;
- 10) усилить кооперацию с организациями стран СЭВ по созданию ПЭВМ и их системного и прикладного обеспечения.

По третьему вопросу повестки дня были заслушаны следующие доклады:

- А. В. Платонов* “О работах по созданию программного обеспечения роботов”,
С. С. Камынин “Программное обеспечение сборочного робота”,
Н. Р. Шведкой “Программное обеспечение сварочного робота”,
В. В. Никифоров “Программное обеспечение устройства УНМ-772 и его развитие”,
С. Л. Зенкевич, А. В. Назарова “Операционная система управления и обучения манипуляционного робота”,
Ю. М. Лазутин “Инструментальное ПО, разрабатываемое в ИПМ АН СССР”.

После обсуждения было принято развернутое решение.

4.11. Одиннадцатое заседание (6–10 сентября 1983 г.)

Очередное заседание Комиссии проходило в сентябре 1983 г. в г. Владивостоке на базе ИАПУ. В повестке дня два вопроса:

1. О работах ИАПУ ДВНЦ АН СССР в области системного программирования.
2. О программном обеспечении машинной графики.

По первому вопросу выступили с докладами:

В. Л. Перчук “О научных исследованиях Института автоматики и процессов управления ДВНЦ АН СССР”,

Е. А. Голенков “О работах ИАПУ в области системного математического обеспечения”,

М. В. Оленин “Системы баз данных в научных исследованиях ИАПУ”,

А. С. Клещев “О работах ИАПУ в области создания экспертных систем”.

Заслушивание и обсуждение докладов по п. 2 повестки дня было предварено вступительным словом А. П. Ершова. Он сказал следующее: “Обсуждать тематические вопросы — традиция Комиссии. Привлекая ведущих специалистов, мы проводим обсуждение состояния работ в СССР с учетом мирового уровня. Цель докладчиков — расширить кругозор членов Комиссии и повысить их способность вырабатывать рекомендации. В ближайшие два года в обсуждении тематических вопросов проявится еще одна составляющая, связанная с намерением перевести управление развитием вычислительной техники на программно-целевой принцип. До этого момента все делали, что могли или что хотели. Теперь стоит задача формулирования конкретных целей для выхода на мировой уровень, т.е. задача быть в состоянии выполнять при наличии ресурсов любую работу. Мы уже свыклись с мыслью об отставании, некоторые организационные формы даже увековечивают это отставание. Программа 12-й пятилетки будет уже строиться на программно-целевом уровне. Мы, как Комиссия, должны дать ответ на вопрос, что, где и в какой срок может быть сделано.

Машграфика из экзотической области программирования становится одной из компонент базового ПО, визуализация становится актуальной в связи с выходом на массового пользователя. В 12-й пятилетке машграфике будет уделяться повышенное внимание, становится важным оценить состояние дел в этой области и дать программу на 12-ю пятилетку.”

Были заслушаны следующие доклады:

- А. А. Карлов, П. В. Вельтмандер, В. А. Бобков, Г. А. Панкеев* “Некоторые вопросы машинной графики: состояние и проблемы”;
- В. А. Бобков* “О работах ИАПУ ДВНЦ АН СССР в области машинной графики”;
- Э. А. Талныкин* “Система синтеза визуальной информации в реальном времени”;
- В. А. Дебелов* “О работах по машинной графике в ВЦ СО АН СССР”;
- Л. Д. Райков, В. В. Городилов, А. А. Мкртумян* “О разработке базисной графической системы в соответствии с проектом международных стандартов ИСО”;
- П. В. Вельтмандер* “Некоторые приложения растровой машинной графики”.

По третьему пункту повестки дня был заслушан и обсужден доклад А. П. Ершова “Пятое поколение ЭВМ”. Докладчик подчеркнул отличие этих машин от ЕС ЭВМ в сторону развития асинхронного типа архитектуры, развитие гетерогенных вычислительных систем. Но каждый процессор внутри системы будет фон-Неймановским. В дискуссии приняли участие М. Р. Шура-Бура, В. Л. Катков, Л. Д. Райков, М. Е. Неменман, И. Я. Ландау, В. М. Курочкин, Э. Х. Тыгу, В. Ф. Хорошевский, И. В. Потосин, Г. Д. Чинин.

4.12. Двенадцатое заседание(17–20 апреля 1984 г.)

Очередное заседание Комиссии проходило в г. Минске на базе Института математики АН БССР и НИИЭВМ. Повестка дня включала следующие вопросы.

1. О разработке программного обеспечения как продукции производственно-технического назначения.

Докладчик — В. И. Саликовский.

2. О работах институтов АН БССР в области системного программного обеспечения.

Докладчики — Н. В. Шкунт, Г. К. Столяров, А. С. Метельский, чл.-корр. АН БССР А. Д. Закревский.

3. О развитии программного обеспечения ЕС ЭВМ.

Докладчики — чл.-корр. АН СССР Г. П. Лопато, Г. В. Пеледов, В. В. Наумов, В. П. Данилочкин, Э. В. Ковалевич, М. П. Котов, В. Л. Катков.

По п. 1 повестки дня был заслушан и обсужден доклад В. И. Саликовского.

Во вступительном слове А. П. Ершов сказал: “В контексте Постановления ЦК КПСС и Совета Министров СССР о развитии работ по вычислительной технике программное обеспечение рассматривается как продукт производственно-технического назначения. Что под этим скрывается? Программное обеспечение — самостоятельное изделие, ПО входит в состав основных фондов предприятия, ПО получает товарную цену, коллективы программистов становится элементом производственной сферы. Этим подводится база (материальная и организационная) под становление отрасли. К этому мы шли давно, и тем не менее постановление застало нас врасплох. Наша дискуссия — это последовательность вопросительных знаков. Руководящие органы задали высокий темп работы по выпуску нормативных документов, подводящих базу под становление отрасли промышленности. Здесь нужно диалектически взять сложившуюся схему, принятую в отраслях, и вложить туда то специфическое, что несет в себе программирование. Это могут сделать только специалисты, нам нужно заняться политэкономией программирования.”

По п.2 повестки дня были заслушаны следующие доклады:

Н. В. Шкут “О работах по математическому обеспечению ЭВМ в Институте математики АН БССР”,

А. Д. Закревский “О работах по математическому обеспечению ЭВМ в Институте технической кибернетики АН БССР”,

Г. К. Столяров “О работах ИМ АН БССР в области СУБД”,

А. С. Метельский “О работе Республиканского ФАП при ИМ АН БССР”.

По п.3 повестки дня были заслушаны следующие доклады:

Г. В. Пеледов “Програмное обеспечение ЕС ЭВМ”,

В. В. Наумов “Операционная система ОС.7”,

В. П. Даниловичин “Програмное обеспечение телеобработки данных ЕС”,

Г. П. Лопато “О структуре и основных направлениях работы НИИ ЭВМ”,

Э. В. Ковалевич “О работах НИИ ЭВМ по программному обеспечению”,

М. П. Котов “Система виртуальных машин ЕС ЭВМ”,

В. Л. Катков “Инструментальный комплекс для разработки ПО ЕС ЭВМ”.

Открывая дискуссию, А. П. Ершов сказал: “Я отмечу ряд моментов, которые не прозвучали в докладах представителей НИЦЭВТа. Наиболее существенное — ничего не было сказано о перспективах, кроме

самых краткосрочных. Реакция на общее развитие ВТ только намечена. Что произвело впечатление? По-прежнему господствует ориентация на пакетную обработку, переход на комбинированный, даже не диалоговый, не прозвучал. Не видно давления на промышленность в части номенклатуры и соотношений. Работы по ЕС развиваются изнутри, наблюдается дефицит суммарных опережающих разработок, отсутствует ряд перспективных направлений. НИЦЭВТ до сих пор не поставляет интегрированные технологии разработки ПО. По обеспеченности оборудованием 10 тыс. ЕС эквивалентны примерно 15 тыс. хороших персональных ЭВМ, т.е. использование ЕС ЭВМ до сих пор нерентабельно.

Очень чувствительным местом является интерфейс, где безответственные научные разработки переходят в серьезные конструкторские решения. Вокруг НИЦЭВТа кормится масса научных организаций, а НИЦЭВТ продолжает всемирный поиск прототипов. Желательно было бы иметь хорошо подготовленных сотрудников, отслеживающих направления, и достаточно авторитетных, чтобы они могли бы что-то внедрить в планы исследовательских работ”.

4.13. Тринадцатое заседание (23–24 ноября 1984 г.)

Очередное заседание Комиссии проходило в г. Ленинграде на базе ИТА АН СССР. В повестку дня было включено 3 вопроса:

1. О работах институтов Ленинградского научного центра в области системного программирования.

Докладчики — чл.-корр. АН СССР С. С. Лавров, Ю. В. Матиясевич, И. В. Клокачев, А. Н. Терехов, Г. С. Цейтин, В. П. Котляров.

2. Доказательное программирование.

Докладчики — чл.-корр. АН ЭССР Э. Х. Тыгуу, В. А. Непомнящий, Я. М. Барздинь.

3. Разное.

По п. 1 повестки дня были заслушаны следующие доклады:

Чл.-корр. АН СССР С. С. Лавров “Работы Института теоретической астрономии АН СССР в области системного программирования.

И. В. Клокачев, Б. Л. Овсевич “О работах Института социально-экономических проблем (ИСЭП) в области системного программирования”,

Ю. В. Матиясевич “О работах ЛОМИ в области системного программирования”,

А. Н. Терехов “О работе лаборатории системного программирования ВЦ ЛГУ”,

Г. С. Цейтин “О работах лаборатории математической лингвистики НИИММ ЛГУ”,

В. П. Котляров “О работах Ленинградского политехнического института им. М. И. Калинина в области системного программирования”.

По п.2 повестки дня были заслушаны следующие доклады:

Э. Х. Тыгу “О трех направлениях в доказательном программировании”,

В. А. Непомнящий, В. К. Сабельфельд “Трансформационный синтез корректных программ”,

Я. М. Барздинь “О состоянии исследований в области индуктивного синтеза программ”.

4.14. Четырнадцатое заседание (27 мая–2 июня 1985 г.)

Очередное, четырнадцатое заседание Комиссии проходило в г. Калинин на базе “Центрпрограммсистем” совместно с Научно-техническими комиссиями ГКНТ СССР по языкам и системам программирования и технологии программирования. Повестка дня включала следующие вопросы:

1. О разработке программного обеспечения как продукции производственно-технического назначения (проекты нормативных документов).
2. О работе ГосФАПа.

По первому вопросу повестки дня были заслушаны и обсуждены доклады В. И. Саликовского о работах, проводимых отделом общесистемного и прикладного программного обеспечения ГУВТиСУ ГКНТ СССР; В. П. Тихомирова об индустрии программного обеспечения; К. Е. Волковицкого о комплексе нормативно-технических документов; В. П. Суханова о деятельности Государственного фонда алгоритмов и программ.

По второму вопросу повестки дня были заслушаны и обсуждены доклады В. П. Тихомирова об основных направлениях деятельности НПО “Центрпрограммсистем”; В. П. Куприянова об опыте разработки и внедрения средств технологии программирования; А.Н.Наумова об основных результатах по разработке и сопровождению программных средств СУБД; Ю. В. Исаева о перспективных работах по программным средствам для ГАП.

4.15. Пятнадцатое заседание (7–10 апреля 1986 г.)

Очередное, пятнадцатое заседание Комиссии проходило в Вильнюсе на базе Института математики и кибернетики АН ЛитССР. Повестка дня включала в себя следующие вопросы:

1. Программное обеспечение персональных ЭВМ.

Докладчики — В. М. Брябрин, М. Е. Неменьман, А. Л. Шмундак, А. В. Гизлавый.

2. О работах Института математики и кибернетики АН ЛитССР в области системного программирования.

Докладчики — А. Л. Телькснис, А. А. Чаплинскас, Р. Й. Шейнаукас, Е. Каргаускас, Г. Григас.

3. Разное.

По первому пункту повестки дня были заслушаны и обсуждены следующие доклады:

В. М. Брябрин “Состояние и проблемы в области персональных ЭВМ”,
М. Е. Неменьман “Персональная ЭВМ ЕС-1840”,
С. И. Сидорас “Новый 32-разрядный вычислительный комплекс”,
А. В. Гизлавый “Состояние разработок учебных персональных ЭВМ”,
А. Л. Шмундак “Система МикроПРИЗ”.

По второму пункту повестки дня были заслушаны и обсуждены следующие доклады:

Л. А. Телькснис “Вычислительная система «Моксиас» для автоматизации научных исследований”,
А. А. Чаплинскас “Инструментальная система «Вильнюс»”,
Й. В. Моцкус “Пакет прикладных программ для глобальной оптимизации”,
Р. Шейнаукас, К. Карчаускас “Программное обеспечение систем автоматизации проектирования”,
Г. К. Григас “Республиканская (заочная) школа молодых программистов”.

Специального решения по этому вопросу не принималось.

5. ЗАВЕРШЕНИЕ РАБОТЫ КОМИССИИ

Четырнадцатое заседание Комиссии в Калининe (май-июнь 1985 г.) проходило без присутствия ее председателя Андрея Петровича Ершова. Он тяжело заболел и был в это же время прооперирован. Как бы по

инерции прошло пятнадцатое заседание в Вильнюсе (1986), где председательствовал А. П. Ершов, намечавшееся шестнадцатое заседание в Киеве не состоялось по независящим от Комиссии причинам. На 1987 г. заседания Комиссии не предусматривалась и планами работы Комиссии предусматривались лишь заседания рабочих групп. Однако, и эта деятельность после смерти А. П. Ершова в декабре 1988 быстро сошла на нет. Назначенный председателем Комиссии по СМО чл.-корр. АН СССР В. П. Иванников произвел кадровые перестройки, в частности был назначен новый ученый секретарь (Дзержинский). Под руководством В. П. Иванникова было проведено еще одно заседание Комиссии по СМО в Новосибирске, но материалы этого заседания остались недо-ступными, если они вообще были подготовлены.

В это же время Председатель ККВТ академик Г. И. Марчук был избран Президентом АН СССР, что полностью отвлекло его от работы ККВТ. В результате деятельность Комитета была прекращена без лишнего шума.

6. РАБОЧИЕ ГРУППЫ КОМИССИИ

При Комиссии по СМО для более детальной проработки отдельных вопросов, представляющих особый интерес для общественности были созданы Рабочие группы по реализации языков программирования (председатель И. В. Поттосин, ученый секретарь Г. Г. Степанов), по языку программирования АЛГОЛ 68, по синтезу и преобразованиям программ (председатель Э. Х. Тыгу, ученый секретарь П. В. Эомойс). Для обсуждения более частных вопросов были созданы при рабочих группах Целевые подгруппы по автоматизации трансляции, по расширяющимся языкам и системам (председатель Д. Н. Тодорой), по аттестации трансляторов и систем программирования (председатель Т. С. Васючкова).

СПИСОК ЛИТЕРАТУРЫ

1. Евстигнеев В. А. О работе Комиссии по системному математическому обеспечению Координационного комитета по вычислительной технике АН СССР // Программирование. — 1984. — N 1. — С.93-94.
2. Евстигнеев В. А. О работе Комиссии по системному математическому обеспечению Координационного комитета по вычислительной технике АН СССР в 1983-1984 гг. // Программирование. 1986. — N 2. — С.93-94.

Г. Г. Степанов

ПРОТОКОЛЫ РАБОЧЕЙ ГРУППЫ ПО РЕАЛИЗАЦИИ ЯЗЫКОВ ПРОГРАММИРОВАНИЯ¹

Протокол 1-го заседания Рабочей группы по реализации языков программирования (РГ РЯП)

11.9.1980 г. в г. Таллинне состоялось первое заседание Рабочей группы по реализации языков программирования (РГ РЯП), созданной при Комиссии СМО Координационного комитета по вычислительной технике при Президиуме АН СССР.

Целью РГ РЯП является неформальная координация работ, ведущихся в данной области, установление постоянных деловых контактов между разработчиками, обсуждение основных ведущихся работ с оценкой опыта и перспективности.

В повестке дня были утверждены следующие вопросы:

1. Тематика РГ РЯП.
2. Стиль и форма работы.
3. План работ на ближайшие заседания группы.

В результате обсуждения была выработана следующая **тематика**

Рабочей Группы:

- Методы трансляции.
- Оптимизация.
- Определение языков, ориентированное на реализацию.
- Системы программирования.
- Автоматизация построения трансляторов.
- Языки программирования и их реализационные аспекты.

Отмечено, что методы трансляции включают также вопросы макрогенерации. В процессе обсуждения подчеркивалась важность вопросов оценки трансляторов, оттранслированных программ и влияния аппаратуры на методы реализации языков.

Решено интересоваться системами программирования, языками программирования — кандидатами на реализацию, анализом ограничений

¹Приведены подлинные документы без сокращений и изменений стиля.

и расширений входных языков реализованных систем программирования лишь как окружением процесса трансляции.

Формой работы РГ РЯП выбраны:

- Научные заседания в течение 2–3 дней, на которых будут заслушиваться подготовленные обзоры, доклады о конкретных разработках, сообщения о научной работе членов РГ, обсуждаться монографии и учебники по программированию.
- Подготовка экспертных оценок по поручению комиссии СМО.
- Взаимный обмен информацией, публикациями и материалами.

Следующее заседание ориентировочно назначено на март 1981 г. В повестку дня заседания вынесены следующие вопросы:

1. Обзор методик трансляторов — Ю. М. Баяковский, Б. Ш. Кауфман, А. Н. Терехов.
2. Обзор измерительных средств, включаемых в рабочую программу — С. С. Лавров.
3. Обсуждение перечня английских терминов и их русских переводных эквивалентов — Д. Б. Подшивалов.
4. Рассмотрение работ по системам программирования на МВК Эльбрус с возможным более детальным рассмотрением транслятора с Фортрана — Г. Д. Чинин.

Руководство группы включает бюро группы, председателя и ученого секретаря.

Председателем группы назначен И. В. Поттосин.

Состав бюро группы: В. М. Курочкин, Э. З. Любимский, И. В. Поттосин.

Ученым секретарем группы назначен Г. Г. Степанов.

На заседании группы отсутствовали: Б. Д. Бабаян, Д. С. Клецев, К. С. Кузьмин, А. А. Летичевский, Г. Д. Чинин, Е. Л. Ющенко (вместо нее присутствовала Л. П. Бабенко).

Кроме членов РГ, на заседании присутствовали С. П. Крицкий и С. Б. Покровский.

Членами группы были представлены письменные соображения по поводу тематики Рабочей группы, учтенные в процессе работы.

Протокол 2-го совещания

Рабочей группы по реализации языков программирования

С 22 по 24 апреля 1981 г. в г. Протвино Московской области состоялось совещание РГ РЯП, на котором были рассмотрены 2 вопроса:

1. Системы программирования для МВК Эльбрус.
2. Методы аттестации трансляторов.

В рамках первого вопроса были заслушаны развернутые доклады основных разработчиков МВК Эльбрус.

В докладе Б. А. Бабаяна были изложены основные принципы архитектуры и операционной системы МВК Эльбрус, при этом особое внимание было уделено тем аспектам, которые способствуют реализации трансляторов и систем программирования.

В своем докладе В. М. Пентковский рассмотрел основной реализационный язык Автокод МВК Эльбрус с точки зрения учета специфики трансляции.

Ю. С. Румянцевым был сделан обзор системы программирования Автокод.

В совместном докладе В. А. Маркова, И. С. Голосова, Г. И. Сердюка, В. В. Окольниковича и С. И. Шелестова были изложены разрабатываемые системы программирования для основных алгоритмических языков Фортран, Алгол 60, Кобол, ПЛ-1. При этом было уделено внимание обеспечению переносимости на МВК Эльбрус существующего на ЕС ЭВМ и БЭСМ-6 матобеспечения.

При обсуждении рассмотренного вопроса выступили: В. В. Луцикович, Л. К. Выханду, А. Н. Терехов, М. Г. Гонца, В. Л. Темов, Е. Л. Ющенко, Б. Ш. Кауфман.

В качестве положительных сторон члены Рабочей группы отметили, что разработка содержит ряд новых черт, таких как новое отношение к динамизму, реализационный язык высокого уровня, обеспечивающий разработку различных компонент системного матобеспечения, сближение архитектуры машины с языком программирования. Участниками совещания отмечались хорошие эксплуатационные характеристики Автокода. Обсуждались возможности того, как повлияет высокий уровень базовых понятий системы на реализацию некоторых аспектов языков программирования. В ответных выступлениях реализаторов систем на конкретных примерах было показано, что реализация выглядит достаточно естественно. В общем положительно оценив динамизм типов, Рабочая группа отметила, что статичность типов иногда больше соответствует характеру задач и дает возможность более полного и своевременного контроля.

Были высказаны пожелания о введении возможности управления механизмом распределения виртуальной памяти пользователей.

Замечания некоторых членов группы вызвали поглощение Автокодом функций языка управления заданиями, однако большинство членов РГ оценило это положительно.

Отмечалась полезность внесения таких же функций и в другие входные языки системы.

Разработчикам систем программирования было рекомендовано более дисциплинированно подойти к обеспечению переносимости программ, в частности, внимательно оценить возможность конверторного подхода.

В целом Рабочая группа высоко оценила разработку, как дающую хорошие результаты для методологии и практики реализации языков программирования.

По второму вопросу с основными сообщениями выступили А. Н. Терехов, Т. С. Васючкова и В. Ш. Кауфман. Кроме того, с сообщениями и обсуждениями выступили В. М. Гуцин, В. И. Цагельский, А. С. Клещев, И. В. Поттосин, А. А. Летичевский, М. Е. Неменман и др.

Т. С. Васючкова сделала доклад о методике сопоставления трансляторов, разработанной Уичманом. Она также привела данные об измерениях основных систем программирования на БЭСМ-6.

А. Н. Терехов привел результаты измерений некоторых трансляторов ЕС ЭВМ, сделанных по методике Уичмана.

В сообщении В. Ш. Кауфмана был изложен возможный набор параметров аттестации трансляторов.

В выступлениях была отмечена принципиальная важность работ по аттестации трансляторов для разработчиков трансляторов и то, что эти работы должны стать одним из основных направлений тематики Рабочей группы. Отмечалась многоаспектность проблемы аттестации трансляторов, необходимость проведения экспериментов по аттестации различных трансляторов в рамках РГ РЯП. Для этого создана временная рабочая подгруппа, целью которой является создание экспериментальной методики аттестации трансляторов и проверка этой методики на трансляторах с Фортрана. Рекомендовалась к использованию методика Уичмана.

В последующих обсуждениях высказывались пожелания о дальнейшей тематике работы РГ:

- стандартизация языков программирования,
- язык АДА,
- языки представления знаний,

- языки параллельного программирования и методы их реализации,
- языки типа Декарт и методы их реализации,
- архитектура ЭВМ и ее влияние на языки программирования,
- микропрограммная реализация языков программирования.

Были высказаны пожелания о более интенсивном обмене информацией, в том числе рабочими записками, между членами РГ и о необходимости некоторой подготовки членов РГ к совещаниям.

Было принято решение 3-е совещание организовать в г. Минске в январе–феврале 1982 г. по следующей тематике:

- Системы программирования ЕС ЭВМ (М.Е. Неменман, В.И. Цагельский).
- Экспериментальная аттестация трансляторов с Фортрана для ЕС ЭВМ (А. Н. Терехов, А. С. Клещев, В. Ш. Кауфман).
- Языки программирования макроконвейерной вычислительной системы и методы их реализации (А. А. Летичевский).

Было решено на 4-ом заседании, которое планируется в г. Новосибирске в ноябре 1982 г., рассмотреть автоматизацию построения трансляторов как основной вопрос.

На нынешнем совещании не удалось рассмотреть вопросы:

- включение в рабочую программу средств ее измерения,
- обсуждение перечня английских терминов и их русских переводных эквивалентов.

Решено вернуться к этим вопросам в ходе следующих совещаний.

На совещании отсутствовали члены РГ РЯП: В.М. Курочкин, Э.З. Любимский, Ю.М. Баяковский, С.С. Лавров, Д.Б. Подшивалов, В.Н. Редько, Э.Х. Тыгу, М.Г. Цуладзе, Г.Д. Чинин.

В качестве наблюдателей присутствовали: А.Л. Александров, А.Н. Бирюков, Т.С. Васючкова, В.Ю. Волконский, Л.К. Выханду, И.С. Голосов, Г.Р. Громов, В.М. Гуцин, Д.Н. Тодорой, В.И. Цагельский, С.М. Шелестов, Н.В. Шкут, А.Л. Шмундак.

Сообщение: в состав Рабочей группы включен Федосеев Александр Иванович.

Дом.: 142284, Протвино Московской области, Молодежный проезд, 3, кв.136.

Сл.: 142284, Протвино Московской области, тел. 20-63 (Серпуховский район).

Информационное письмо

Глубокоуважаемый

Сообщаю Вам, что следующее заседание Рабочей группы по методам реализации языков программирования состоится в г. Минске с 1 по 5 февраля 1982 г. Заседание организует НИИ ЭВМ, ответственным за его проведение от Рабочей группы является М. Е. Неменман.

Программа заседания включает следующие вопросы:

1. Системы программирования ЕС ЭВМ и методы реализации языков программирования в этих системах.
 - 1.1. Штатные системы программирования ЕС ЭВМ (вопрос готовится сотрудниками НИИ ЭВМ).
 - 1.1.1. Обзор систем программирования для ЕС ЭВМ.
 - 1.1.2. Методы оптимизации в системах программирования для ЕС ЭВМ.
 - 1.1.3. Встроенные методы отладки в системах программирования для ЕС ЭВМ.
 - 1.2. Новые разработки по системам программирования для ЕС ЭВМ.
 - 1.2.1. Система программирования для Алгол 68 (А. Н. Терехов, Г. С. Цейтин — ЛГУ).
 - 1.2.2. Система программирования для Алгамса (Н. В. Шкут — ИМ АН БССР, Б. А. Кондратенко — ВЦ РГУ).
 - 1.2.3. Диалоговая система для ПЛ/1 (Р. П. Балодис — ЛатГУ).
 - 1.2.4. Специализированная система программирования для ЕС ЭВМ (С. С. Гороховский, В. В. Бублик — ИК АН УССР).
 - 1.2.5. Представление систем программирования для Паскаля (В. Н. Касьянов — ВЦ СО АН СССР).
2. Методика аттестации трансляторов (отв. — В. Ш. Кауфман, А. С. Клещев, А. Н. Терехов).
3. Математическое обеспечение макроконвейерной вычислительной машины (А. А. Летичевский — ИК УССР).

Во время заседаний также предполагается уточнить ближайшие планы Рабочей группы (напоминаю, что по предварительным наметкам следующее заседание Рабочей группы предполагается провести в ноябре 1982 г. в г. Новосибирске с основной темой — автоматизация построения трансляторов).

Ваши замечания по программе заседаний и по планам группы прошу сообщить до конца декабря 1981 г. или немедленно, в зависимости от

характера замечаний, для реальной возможности учесть их.

Прошу учесть в Ваших планах то, что ввиду большой программы заседание будет проходить полностью 3 дня (2, 3 и 4 февраля). Предполагается, что предложения по методике аттестации трансляторов, подготовленные ответственными за этот вопрос, будут предварительно рассмотрены членам Рабочей группы.

Для подготовки третьего вопроса члены Рабочей группы могут ознакомиться со статьями В. М. Глушкова, Ю. В. Капитоновой и А. А. Летищевского — “О построении семейства алгоритмических языков для программирования и проектирования многопроцессорных вычислительных систем” (“Кибернетика”, N 1, 1981 г.), “Об одном подходе к реализации параллельных вычислений в многопроцессорных вычислительных системах” (в сб. “Параллельное программирование и вычислительные системы”, ч. 1, Новосибирск, 1980 г.).

Официальное приглашение и дополнительная информация будут высланы Комиссией по системному математическому обеспечению Координационного комитета по вычислительной технике АН СССР.

Председатель РГ РЯП
И. В. Поттосин

Протокол 3-го заседания Рабочей группы по реализации языков программирования

С 26 по 28 января 1982 г. в г. Минске проходило очередное заседание Рабочей группы по реализации языков программирования (РГ РЯП). На повестке дня было 3 вопроса:

1. Системы программирования ЕС ЭВМ (штатные и оригинальные).
2. Методика аттестации трансляторов.
3. Математическое обеспечение макроконвейерной вычислительной машины.

По первому вопросу представители НИИ ЭВМ В. И. Цагельский, В. Ф. Борисевич, Ф. И. Ефременков, В. И. Воюш, З. С. Брич, С. И. Балацкова, Л. М. Романовская, Т. А. Михальченко, А. Т. Федоров, И. И. Пилецкий подробно, насколько позволяло ограниченное время проведения группы, познакомили членов РГ со штатным математическим обеспечением ЕС ЭВМ и математическим обеспечением, разрабатываемым в НИИ ЭВМ.

Были рассмотрены оптимизирующий и отладочный трансляторы с ПЛ/1, оптимизирующий транслятор с Фортрана, система ДИОП, средства отладки, в том числе и отладки в режиме разделения времени.

В конце В. И. Цагельский и А. С. Марков рассказали о перспективах дальнейшего развития матобеспечения ЕС ЭВМ. Выступили А. Н. Терехов (Алгол 68), А. М. Колесник и В. А. Кондратенко (о разрабатываемых в Минске и Ростове трансляторах с Алгамса), Р. П. Балодис (о диалоговой системе для подмножества ПЛ/1 с самостоятельными канальными программами), С. С. Гороховский (специализированная система программирования КЛУБОК), В. Н. Касьянов (о двух системах с Паскаля, разработанных в Польше и Чехословакии) и В. М. Малышев (о разработанном в НИЦЭВТе трансляторе реального времени с Фортрана).

В ходе состоявшейся дискуссии выступали Л. Ф. Белоус, М. Г. Гонца, В. Ш. Кауфман, А. С. Клещев, Э. В. Ковалевич, А. С. Марков, Т. А. Михальченко, М. Е. Неменман, И. В. Потгосин, А. Н. Терехов, А. Т. Федоров, В. И. Цагельский, Г. С. Цейтин, М. Р. Шура-Бура. Рассматривались, в основном, вопросы о языке общения пользователей с матобеспечением, о способе создания промышленных продуктов и о возможности и необходимости внесения изменений в адаптируемое матобеспечение.

Учитывая процессы адаптации и используемость матобеспечения ЕС ЭВМ в странах СЭВ, РГ тем не менее считает необходимым для массового пользователя в нашей стране наличие версий трансляторов с русской диагностикой. Отмечалась большая роль НИИ ЭВМ и НИЦЭВТа в разработке промышленного матобеспечения ЕС ЭВМ. Поднимались вопросы доведения до промышленного уровня нестандартных разработок. Высказывалось мнение, что многие "дикие" программы и не нуждаются в полном объеме документации, требуемой для промышленного продукта, и должна быть классификация программ по степени документуемости.

Вопрос о внесении изменений в адаптируемое матобеспечение очень трудный. Что касается разработок конверторов и других вспомогательных программ, то, конечно, их можно только приветствовать. Однако при внесении изменений в алгоритм адаптируемых трансляторов следует учитывать не только полезность вносимых изменений, но и ту цену, которую приходится платить, и уменьшение надежности получаемого продукта.

В рамках подготовки ко второму вопросу Рабочей подгруппой в со-

ставе В. Ш. Кауфмана, А. С. Клещева, А. Н. Терехова был выпущен и разослан членам РГ материал “О методике аттестации трансляторов”. Следует отметить, что в ходе работы подгруппы к ней присоединился А. Л. Александров, а в ходе работы совещания — А. С. Марков. С сообщением от Рабочей подгруппы выступил А. Н. Терехов. Подгруппой тестировался транслятор Фортран ОС ЕС ЭВМ. Частью аттестации была оценка транслятора по методике Уичмана. А. Н. Терехов рассказал о замеченных недостатках аттестуемого транслятора и поблагодарил коллектив разработчиков за создание благоприятной обстановки для работы подгруппы.

Затем Т. С. Васючкова выступила с сообщением о различных методиках оценки характеристик трансляторов, моделях и формулах сравнительных оценок качества выходного кода трансляторов.

В обсуждении докладов и дискуссии приняли участие С. Н. Берестовая, М. Г. Гонца, В. Н. Касьянов, В. Ш. Кауфман, А. С. Клещев, Э. В. Ковалевич, В. М. Курочкин, М. С. Марголин, А. С. Марков, И. И. Пилецкий, И. В. Поттосин, В. Л. Темов, А. Н. Терехов, А. Т. Федоров, М. Н. Шура-Бура.

Были подняты вопросы для дальнейшей работы в области аттестации трансляторов и систем программирования:

- расширение смеси Уичмана для современных языков,
- проверка правильности транслятора,
- накладные расходы ОС,
- аттестация языков,
- аттестация диагностики,
- аттестация средств отладки,
- аттестация документации и сопровождения,
- оценка легкости общения с системой,
- аттестация многоязыковых трансляторов,
- аттестация комплекса взаимодополняющих трансляторов,
- необходимость создания банка тестов языков программирования,
- методика Холстеда.

Этот внушительный перечень поднятых вопросов говорит о большой важности работ по аттестации трансляторов систем программирования. Отмечалось, что следует различать аттестации, ориентированные на разработчиков, и аттестации, ориентированные на пользователей. Но и пользователю нужна не интегрированная оценка транслятора, а набор характеристик.

РГ просила сотрудников НИИ ЭВМ прислать членам подгруппы по аттестации свои письменные замечания, после чего выпустить новый текст материала “О методике аттестации трансляторов”.

РГ призывает членов подгрупп и других своих членов продолжать активные работы по аттестации трансляторов.

По третьему вопросу с обширным докладом выступил А. А. Летичевский, в котором рассказал о разработке рекурсивной вычислительной машины на базе машин серии ЕС ЭВМ и языка программирования для нее — МАЯК. В ходе обсуждения А. А. Летичевский сообщил, что ведутся также работы по моделированию разрабатываемой машины и матобеспечения, а также работы по автоматическому распараллеливанию алгоритмов. Срок выхода серийных экземпляров машины — 1985 г.

Основной темой следующего совещания РГ выбрана автоматизация построения трансляторов. Вместо предполагавшегося места проведения совещания (ноябрь 1982 г. в Новосибирске) решено провести его под Москвой в декабре 1982 г.

На совещании отсутствовали члены РГ РЯП: Э. З. Любимский, Б. А. Бабаян, С. С. Лавров, В. М. Пентковский, Д. Б. Подшивалов, В. Н. Редько, Э. Х. Тыгу, А. И. Федосеев, М. Г. Цуладзе, Г. Д. Чинин, Е. Л. Ющенко.

В качестве наблюдателей присутствовали: А. Л. Александров, Р. П. Балодис, С. Н. Берестовая, В. И. Билан, А. Н. Бирюков, В. В. Бублик, Т. С. Васючкова, И. С. Голосов, С. С. Гороховский, В. А. Евстигнеев, Б. П. Ершов, В. А. Кондратенко, Л. А. Корнева, В. М. Малышев, В. В. Островский, Д. Н. Тодорой, Г. С. Цейтин, А. Л. Шмундак, М. Р. Шура-Бура, сотрудники НИИ ЭВМ.

Информационное письмо

Уважаемый

Сообщаю Вам, что предварительное время проведения очередного заседания Рабочей группы по реализации языков программирования — с 30 ноября по 4 декабря 1982 г., место проведения — под Москвой.

Планируемая программа заседания:

1. Научные доклады о системах построения трансляторов:
 - СПТ РГУ (докладчик В. П. Пыхалов),
 - СПТ СУПЕР (докладчик В. А. Серебряков),
 - СПТ ELMA (докладчик А. О. Вооглайд),
 - СПТ ТУ (докладчик В. В. Луцикович),
 - СПТ КРОСС (докладчик В. И. Гололобов),

- система МАСОН (*докладчик В. Л. Темов*),
 - возможности системы ПРИЗ как СПТ (*докладчик Я. Э. Пеньям*).
2. Научные доклады о применении модульного подхода для построения трансляторов:
- опыт системы БЕТА (*докладчик С. Б. Покровский*),
 - опыт системы Алгол 68 (*докладчик А. Н. Терехов*),
 - опыт системы РГУ (*докладчик Х. Д. Дженибалаев*).
3. Общая дискуссия о методах автоматизации построения трансляторов.

Точное время проведения и дополнительная информация об организации заседания будут указаны в приглашении, которое будет разослано Комиссией по системному математическому обеспечению ККВТ АН СССР.

Ученый секретарь РГ РЯП
Г. Г. Степанов

Протокол 4-го заседания Рабочей группы по реализации языков программирования

С 20 по 22 ноября 1982 года в г. Протвино состоялось очередное заседание РГ РЯП.

На повестке дня были следующие вопросы:

1. Модульный подход к построению трансляторов.
2. Системы построения трансляторов (СПТ).
3. План работы на 1983 год.

20 ноября было посвящено рассмотрению первого вопроса. С докладами выступили:

С. Б. Покровский “Опыт системы БЕТА”.

И. Б. Гиндыш, А. Н. Терехов “Опыт системы Алгол 68”.

С. М. Абрамович “Опыт системы РГУ”.

Г. С. Цейтин “Организация модульности на основе сетевых структур”.

Второй вопрос обсуждался весь день 21 ноября и первую половину 24 ноября. С докладами выступили:

В. Ш. Кауфман “Представление знаний в СПТ”.

В. П. Пухалов “СПТ РГУ”.

В. А. Серебряков “СПТ СУПЕР”.

А. О. Вооглайд, М. В. Лепп, М. Б. Меристе “СПТ ЕЛМА”.

В. В. Луцикович “СПТ ТУ”.

В. Л. Темов “Система МАСОН”.

Я. Э. Пеньям “Возможности системы ПРИЗ как СПТ”.

Кроме того, А. А. Летичевский сделал сообщение о разрабатываемой в ИК АН УССР системе ТЕРЕМ.

При обсуждении докладов о модульном подходе к автоматизации построения трансляторов было отмечено, что уже существуют проверенные и оправдавшие подход эксперименты с построением этапа декомпозиции (система БЕТА) и этапа генерации (системы на базе Алгола 68) на основе такого подхода.

Вместе с тем использование уже разработанных в этих системах библиотек процедур (в их существующей реализации) в других системах невозможно в силу ориентации только на свойства тех систем, в рамках которых эти библиотеки создавались. Более общий подход к модульному анализу процессов трансляции и к самому выделению библиотек стандартных модулей трансляции мог бы дать некоторые универсальные инструменты для создателей трансляторов. Дальнейшее движение в этом направлении представляет проект, планы которого существуют в РГУ, однако его цели показались некоторым членам РГ слишком всеобъемлющими, и, возможно, для успеха проекта следует более четко ограничить эти цели.

Некоторые эксперименты, проведенные в ЛГУ, говорят о возможности фиксации модульности программных систем в виде сетевых структур.

В докладах по СПТ были представлены основные советские работы в этом направлении. Ряд реализованных СПТ превращается в развитые инструментальные системы для построения языковых процессоров — СПТ ELMA, МАСОН. Имеется опыт по созданию специализированных языковых процессоров на основе СПТ ELMA, СПТ РГУ, предназначенных для промышленного производства трансляторов с универсальных языков.

Весьма перспективной инструментальной системой является система МАСОН, ценность которой заключается в том, что она дает один из подходов к созданию программного обеспечения персональных ЭВМ, ориентированных на системного программиста.

В проекте системы ТУ важным является ориентация на формальное описание выходной машины.

Большой методологический и теоретический интерес представляют исследования атрибутивного подхода к построению трансляторов, осу-

ществляемые в рамках систем СУПЕР и ELMA. Использование системы ПРИЗ может дать инструмент для экспериментов с семантическими описаниями языков программирования. Проекционный подход уже был применен для автоматизации построения трансляторов со специализированных языков.

В целом, все обсуждаемые работы представляются перспективными и взаимодополняющими. При обсуждении высказывалось мнение о внимательном изучении других подходов к построению СПТ, более широком сопоставлении формализмов описания языков и их возможностей для автоматизации построения трансляторов (в будущем возможно создание соответствующей целевой подгруппы).

В ряде систем уже сейчас имеются некоторые фрагменты, которые можно было бы использовать и вне этих систем при соответствующем их оформлении разработчиками.

Более широкому внедрению систем мешает отсутствие документации, руководств и т.п., для создания чего следовало бы привлечь организации, ответственные за внедрение программного обеспечения, и, возможно, вузы. Технологические подходы к созданию трансляторов, например, расслоенное программирование, было бы полезно подкреплять инструментально и соответствующим языковым оформлением.

Была высказана рекомендация разработчикам системы ELMA предоставить участникам Таллиннской конференции возможность более глубоко познакомиться с системой и провести сеансы работы с ней. Для рассмотрения системы ТЕРЕМ (она излагалась только в кратком сообщении А. А. Летичевского и вызвала общий интерес) было бы целесообразно поставить доклад о ней на следующем заседании РГ.

После общей дискуссии были выработаны следующие решения заседания РГ РЯП:

1. Настоятельно рекомендовать разработчикам систем автоматизации построения трансляторов оформлять отдельные части своих разработок в виде программного продукта, готового к использованию при разработке других систем.
2. Рекомендовать разработчикам СПТ уделять внимание выработке общих методик построения СПТ.
3. Организовать Целевую подгруппу по измерению систем программирования.
4. Обратить внимание на вопросы тестирования СПТ и аттестации систем программирования.

5. Просить Комиссию ГКНТ по языкам и системам программирования организовать аттестацию системы МАСОН с целью определения перспектив ее развития и практического использования.
6. Просить НИЦЭВТ включить систему МАСОН в состав базового программного обеспечения ЕС ЭВМ и оказать для этого необходимое содействие разработчикам.

Был принят следующий план мероприятий на 1983 год:

1. Провести одно заседание РГ РЯП в г. Кишиневе с 3 по 5 июня, на котором необходимо рассмотреть следующие вопросы:
 - расширяемые языки и системы (отв. Д. Н. Тодорой и Л. Ф. Белоус);
 - системное окружение языковых процессоров (отв. В. М. Пентковский).
2. Провести совместный семинар с Целевой подгруппой по технологии программирования микропроцессорной техники РГ ТП по языкам и методам их реализации для микроЭВМ в г. Новосибирске 27 июня (отв. И. В. Поттосин и Р. К. Рауд).
3. Провести рабочий семинар реализаторов языка Симула-67 в г. Новосибирске (отв. С. М. Абрамович и К. С. Кузьмин).
4. Создать Целевую подгруппу по измерению систем программирования в составе: Т. С. Васючкова (председатель), А. Л. Александров, Ю. М. Баяковский, В. Ш. Кауфман, А. С. Клещев, А. С. Марков, М. Е. Неменман, А. Н. Терехов, В. И. Цагельский. Направлениями работ подгруппы являются создание методик измерения характеристик систем программирования, организация банка данных измеренных характеристик, анализ измерений и рекомендации по использованию трансляторов. Поручить ЦПГ ИСП к концу года подготовить перечень характеристик трансляторов и СПТ и методов их измерения.
5. Подготовить создание ЦПГ по расширяемым языкам и системам (отв. Д. Н. Тодорой).
6. Рекомендовать членам РГ РЯП принять активное участие и оказать содействие в проведении совещаний в г. Таллинне (март) и в г. Кишиневе (май–июнь).

На заседании отсутствовали: В. М. Курочкин, Б. А. Бабаян, Ю. М. Баяковский, В. Н. Касьянов, А. С. Клещев, В. М. Пентковский, Д. Б. Подшивалов, В. Н. Редько, Э. Х. Тыгу, М. Г. Цуладзе, Г. Д. Чинин.

В качестве наблюдателей присутствовали: А. Л. Александров, Т. С. Васючкова, И. Б. Гиндыш, Х. Д. Дженибалаев, В. А. Евстигнеев, О. А. Ильичева, С. П. Крицкий, М. В. Лепп, В. П. Макаров, Я. Э. Пеньям, С. Б. Покровский, В. Л. Пыхалов, Д. Н. Тодорой, В. И. Цагельский, Г. С. Цейтин, М. Р. Шура-Бура.

Рабочая группа пополнена следующими членами:

Васючкова Татьяна Сергеевна

сл.: 630090, г. Новосибирск, пр. Лаврентьева, 6, НФ ИТМ и ВТ,
тел. 65-11-23

дом.: 630090, г. Новосибирск, ул. Ильича, 19, кв. 38.

Тодорой Дмитрий Николаевич

сл.: 277003, г. Кишинев, ул. Садовая, 60, Госуниверситет,
Кафедра алгоритмических языков программирования,
тел.: 25-00-21 доб. 5-94 или 5-98

дом.: 277044, г. Кишинев, ул. Кирова, 4, корпус 1, кв. 301,
тел. 433-07-02.

Цагельский Владимир Иосифович

сл.: 220600, г. Минск, НИИ ЭВМ, тел. 34-53-84

дом.: 220090, г. Минск, ул. Я. Коласа, 139, кв. 106.

Просьба ко всем членам группы включить указанных товарищей в списки рассылки материалов.

В разосланном ранее списке адресов и телефонов членов группы произошли следующие изменения:

В. Л. Темов

место работы: г. Ленинград, ПО ЛЭМВ ВНИКИ ОУПУ,
тел. 247-89-25.

Д. А. Бухштаб

сл.тел.: 139-10-22 или 139-24-71.

В. М. Курочкин

дом.: 117421, г. Москва, ул. Новаторов, 40, корпус 2, кв. 49,
тел. 133-43-03.

Г. Г. Степанов

дом.: 630055, г. Новосибирск, ул. Полевая, 8, кв. 85.

Отчет о деятельности за 1982 г.

Рабочей группы по реализации языков программирования,

действующей при Комиссии по системному математическому обеспечению Координационного комитета по вычислительной технике АН СССР

В 1982 г. проведены следующие мероприятия:

1. Проведено заседание группы в г. Минске — 42 чел.
2. Прделана подготовительная работа по проведению совместного семинара с Целевой подгруппой по технологии программирования микропроцессорной техники РГ ТП по языкам и методам их реализации для микроЭВМ в 1983 г.
3. Подготовлено создание Целевой подгруппы по измерению систем программирования, которая, не будучи оформлена официально, уже проделала большую работу по измерениям характеристик трансляторов на ЕС ЭВМ и БЭСМ-6.
4. Подготовлен вопрос научной программы для заседания комиссии ГКНТ по языкам и системам программирования, посвященный методикам аттестации трансляторов и систем программирования.
5. Проведено заседание группы в г. Протвино — 37 чел.

Информация о проведенных заседаниях группы:

1. С 26 по 28 января в г. Минске проходило очередное заседание группы по следующим вопросам:
 - Системы программирования ЕС ЭВМ, штатные и оригинальные.
 - Методика аттестации трансляторов.
 - Математическое обеспечение макроконвейерной вычислительной машины.
 - Были заслушаны 10 докладов по базовым системам ЕС ЭВМ и 7 докладов по отечественным разработкам.
 - Проведено обсуждение докладов и общая дискуссия, в ходе которой отмечалась необходимость наличия в нашей стране версий трансляторов с русской диагностикой.
 - Рассматривались вопросы о доведении до промышленного уровня нестандартных разработок и вопрос о необходимости внесения изменений в адаптируемое матобеспечение.
 - Рассматривались различные методики измерения характеристик трансляторов и некоторые конкретные результаты измерений, проведенные на ЕС ЭВМ и БЭСМ-6. Отмечалась важность данной тематики.
 - Принято решение о создании Целевой подгруппы.
 - Заслушан доклад А. А. Летичевского о разработке рекурсивной вычислительной машины на базе машин серии ЕС ЭВМ и языка программирования для нее — МАЯК.

2. С 20 по 22 ноября в г. Протвино проходило очередное заседание группы по следующим вопросам:

- Модульный подход к построению трансляторов (СПТ).
- Системы построения трансляторов (СПТ).
- План работы на 1983 г.

Были заслушаны и обсуждены 11 докладов по основным советским разработкам в области автоматизации построения трансляторов. После общей дискуссии было решено:

- настоятельно рекомендовать разработчикам систем автоматизации построения трансляторов оформлять отдельные части своих разработок в виде программного продукта, готового к использованию при разработке других систем;
- рекомендовать разработчикам СПТ уделять внимание выработке общих методик построения СПТ;
- обратить внимание на вопросы тестирования СПТ и аттестации систем программирования;
- просить комиссию ГКНТ по языкам и системам программирования организовать аттестацию системы МАСОН с целью определения перспектив ее развития и практического использования;
- просить НИЦЭВТ включить систему МАСОН в состав базового программного обеспечения ЕС ЭВМ и оказать для этого необходимое содействие разработчикам.

Рабочая группа пополнена следующими членами: Т. С. Васючкова (НФ ИТМ и ВТ, г. Новосибирск), Д. Н. Тодорой (КГУ, г. Кишинев) и В. И. Цагельский (НИИ ЭВМ, г. Минск).

План мероприятий на 1983 г.

Рабочей группы по реализации языков программирования

1. Провести одно заседание РГ РЯП в г. Кишиневе с 3 по 5 июня, на котором рассмотреть следующие вопросы:
 - расширяемые языки и системы (отв. Д. Н. Тодорой и Л. Ф. Белоус);
 - системное окружение языков процессоров (отв. В. М. Пентковский).
2. Провести совместный семинар с Целевой подгруппой по технологии программирования микропроцессорной техники РГ ТП по языкам и методам их реализации для микроЭВМ в г. Новосибирске 27 июня (отв. И. В. Поттосин и Р. К. Рауд).

3. Провести рабочий семинар реализаторов языка Симула-67 в г. Новосибирске (отв. С. М. Абрамович и К. С. Кузьмин).
4. Создать Целевую подгруппу по измерению систем программирования в составе: Т. С. Васючкова (председатель), А. Л. Александров, Ю. М. Баяковский, В. Ш. Кауфман, А. С. Клещев, А. С. Марков, М. Е. Неменман, А. Н. Терехов, В. И. Цагельский. Направлениями работ подгруппы являются создание методик измерения характеристик систем программирования, организация банка данных измеренных систем программирования, организация банка данных измеренных характеристик, анализ измерений и рекомендации по использованию трансляторов. Поручить ЦПГ ИСП к концу года подготовить перечень характеристик трансляторов и СПТ и методов их измерения.
5. Подготовить создание ЦПГ по расширяемым языкам и системам (отв. Д. Н. Тодорой).
6. Рекомендовать членам РГ РЯП принять активное участие и оказать содействие в проведении конференций в г. Таллинне (март) и в г. Кишиневе (май-июнь).

Информационное письмо

Члену (наблюдателю) РГ РЯП Комиссии СМО 1983 г.

Глубокоуважаемый

Комиссия системного математического обеспечения Координационного Комитета по Вычислительной технике АН СССР приглашает Вас принять участие в заседании Рабочей группы по реализации языков программирования, которое состоится с 3 по 5 июня 1983 г. в г. Кишиневе сразу после 4-го Всесоюзного симпозиума "Системное и теоретическое программирование".

Основная тема заседания: расширяемые языки и системы программирования. Подробная программа прилагается.

Командировки выписываются на Кишиневский госуниверситет. Приглашение не подлежит передаче другому лицу.

За справками обращаться к секретарю Комиссии СМО Бухштаб Дине Абрамовне (междугородний тел. 8-095-139-10-22) или к секретарю РГ РЯП Степанову Георгию Георгиевичу (междугородний тел. 8-383-2-65-11-53).

Председатель Комиссии СМО
член-корр. АН СССР А. П. Ершов

Программа работы 5-го заседания Рабочей группы по реализации языков программирования

Заседание РГ РЯП будет происходить в г. Кишиневе с 3 по 5 июня 1983 г.

Основная тема заседания: расширяемые языки и системы программирования.

Планируется следующая программа заседания:

3 июня:

1. *Д. Б. Подшивалов* “О терминологии в языках и методах трансляции” (40 мин).
2. *А. А. Летичевский, Н. М. Мищенко* “Реализация языка Маяк” (1 час).
3. *С. С. Лавров* “Расширяемость языков: теория и практика” (40 мин).
4. *Д. Н. Тодорой* “Расширяемые языки и системы — способы их реализации” (40 мин).
5. *Г. С. Цейтин* “Абстрактные типы данных и механизмы расширения в языках ALPHARD, CLU, ADA” (40 мин).

4 июня:

1. *Ю. В. Капитонова, А. А. Летичевский, Н. М. Мищенко* “Расширяемые языки программирования системы Проект и способы их реализации” (40 мин).
2. *В. М. Пентковский, Б. П. Синдеев* “Об автоматизации создания проблемных языков и окружений на базе универсального языка и системы” (40 мин).
3. *А. О. Вооглайд, М. Г. Рябовейтра* “Расширяемость и переносимость в языке FORIN” (40 мин).
4. *Л. Ф. Белоус, Ю. В. Пастухов* “SYM — синтаксически управляемый макропроцессор общего назначения” (40 мин).
5. *М. И. Селюн, Е. Н. Капустина* “Система АБВ” (40 мин).

5 июня:

1. *А. Г. Красовский* “О роли метаязыка в проекте ИСКРА” (20 мин).
2. *А. С. Клещев* “Фреймовые модели и фрейм-ориентированная система программирования” (20 мин).
3. *А. Л. Шмундак* “МИС — мобильная инструментальная система разработки пакетов программ” (40 мин).
4. *В. А. Серебряков* “Послойное описание процесса трансляции в СПТ СУПЕР и вопросы мобильности” (20 мин).

5. *В. П. Бабенко, Л. А. Мельник, В. В. Синяковская* “Особенности средств генерации прикладных программ в среде СУБД” (20 мин).
6. *Ю. В. Пастухов* “Сравнительные характеристики макропроцессоров общего назначения для ЕС ЭВМ” (20 мин).
7. *С. В. Вдовкин, А. А. Кубанский, В. Ф. Сафонов* “Реализация языка CLU” (40 мин).
8. *М. С. Марголин* “Повышение надежности макроопределений” (40 мин).

Протокол 5-го заседания Рабочей группы по реализации языков программирования

С 3 по 5 июня 1983 г. в Кишиневском госуниверситете проходило 5-е заседание Рабочей группы по реализации языков программирования.

Основным вопросом повестки дня был вопрос о расширяемых языках и системах. В докладах и дискуссиях были обсуждены как общие проблемы расширяемых языков и систем, так и опыт конкретной реализации таких систем.

Было отмечено разнообразие механизмов расширений, применяемых в языках и системах программирования. К средствам расширения можно отнести определение процедур, определение типов, макротехнику, введение абстрактных и инкапсулированных типов данных, средства синтеза программ.

В докладе С. С. Лаврова (Ленинград) были рассмотрены проблемы расширяемых языков и систем на примерах расширений разного уровня. В качестве примеров приводились языки АБВ и Декарт. На их основе демонстрировалась реализация таких проблем расширяемости, как синтаксическая и семантическая расширяемость, выбор базовых конструкций, наличие внешних и внутренних средств расширения, эффективная реализация расширения.

В докладе Д. Н. Тодорова (Кишинев) предлагалась классификация средств расширения, методов их реализации, уровень включения расширения и общая модель расширяемой системы.

В докладе В. С. Сафонова (Ленинград) был дан обзор средств расширения, предлагаемых языками с абстрактными типами данных (АТД), излагались проблемы, возникающие в языках в связи с включением АТД, и трудности реализации, существующие для таких языков. Отмечалась недостаточность средств спецификации АТД в реальных языках.

В докладе А. С. Клещева (Владивосток) вопросы расширяемости и адаптации к проблемам области рассматривались в связи с созданием экспертных систем и излагался опыт создания теоретических основ, технологии и технологической поддержки разработки этих систем на базе фреймовых моделей и фрейм-ориентированной системы программирования.

В. А. Серебряков (Москва) докладывал о проблемах расширения и мобильности в атрибутных системах построения трансляторов (СПТ) и отмечал ряд таких важных проблем реализации этих систем, как необходимость послыного описания входных языков и объединения СПТ и СУБД.

Необходимость объединения СПТ с расширяемой системой отмечалась в докладе А. А. Летичевского и Н. М. Мищенко (Киев), и возможность такого объединения демонстрировалась на примере системы программирования ТЕРЕМ. Авторы отмечали, что при создании базового языка с возможностями расширения минимальность базового языка может быть компенсирована мощными возможностями расширений.

В докладе В. М. Пентковского и Б. П. Синдеева (Москва) было рассказано об автоматизации создания проблемных языков окружения на базе универсального языка и системы за счет использования в основном такого средства расширения, как синтаксические макросы.

Как один из перспективных подходов к созданию специализированных языков путем расширения, в докладе А. О. Вооглайда и М. Г. Рябовейтра (Таллинн) предлагался подход, основанный на языке ФОРТ. Была отмечена возможность использования языка ФОРТ как промежуточного языка, рассмотрена проблема мобильности ФОРТсистем и одновременно отмечалось такое свойство языка ФОРТ, как простота реализации в ущерб надежности.

Э. Х. Тыгу (Таллинн) отметил для таких средств расширения, как АТД, необходимость задания спецификации и реализации АТД, важность введения взаимодействия типов (сочетания, дополнения и т.д.), сказал, как эти возможности реализованы в такой системе синтеза программ, как ПРИЗ.

В докладе В. Н. Капустиной и М. И. Селюна (Москва) был изложен опыт реализации языка АБВ и было показано, как базовые средства языка позволяют реализовать сложные синтаксические конструкции.

А. А. Кубанский (Ленинград) рассказал о методах реализации языка CLU (в первую очередь, средств расширения) в трансляторе для МВК

Эльбрус.

А. Л. Шмундаком (Таллинн) была представлена система МИС — мобильная инструментальная система для создания ППП, являющаяся развитием систем ПРИЗ и уделяющая особое внимание эффективности реализации.

А. Г. Красовский (Москва) рассказал о системе ИСКРА (Интегрированная Система Качественной Разработки) с базовым языком Рефал, отметил, в частности, возможности верификации программ на расширении Рефала, реализованные в системе.

Л. Ф. Белоус и Ю. В. Петухов рассказали о синтаксически управляемом макропроцессоре общего назначения и привели некоторые сравнительные характеристики нескольких макропроцессоров для ЕС ЭВМ.

На заседании Рабочей группы была отмечена важность работ по расширяемым языкам и системам, принципиальная значимость средств расширения для языков и систем программирования, необходимость анализа направления ведущихся работ. Было выделено два конкретных направления — анализ и классификация средств расширения в языках программирования, исследование характеристик существующих макрогенераторов.

Для совместной работы по этим направлениям было принято решение о целесообразности создания двух соответствующих целевых подгрупп, в связи с чем ряду членов и наблюдателей РГ было дано задание подготовить к следующему заседанию РГ уточнение направлений деятельности и конкретные цели каждой ЦПГ.

Предполагаемый состав первой ЦПГ: Д. Н. Тодорой (председатель), Н. М. Мищенко, М. И. Селюн.

Предполагаемый состав второй ЦПГ: Л. Ф. Белоус, М. С. Марголин, А. С. Марков, И. И. Пилецкий, В. Л. Темов.

Помимо основного вопроса на заседании РГ были рассмотрены проблемы терминологии в области языков программирования и методов трансляции. В докладе Д. Б. Подшивалова (Москва) и последовавшей дискуссии отмечался ряд серьезных затруднений, связанных с отсутствием целенаправленной деятельности в этом направлении — неограничиваемое влияние англоязычных плохо переведенных терминов, несогласованность и случайность выбора терминологии, наличие различных терминов для одинаковых или близких понятий.

Отмечались связанные с этим проблемы преподавания, оформления документации и т.п. Была организована Целевая подгруппа по терми-

нологии (Д. Б. Подшивалов — председатель, В. М. Курочкин, С. С. Лавров, А. С. Марков, С. Б. Покровский), задачей которой является подготовка комментированного реестра терминов в соответствии с определяемыми понятиями (своего рода терминологического словаря), выработка предварительных соображений по отбору терминов и методика терминологической работы в области деятельности РГ.

Стандартизация не является непосредственной задачей ЦПГ, ее работа должна заключаться в анализе и комментировании терминов и привлечении внимания общественности к составу существующей терминологии. РГ приняла решение обратиться к редколлегии журнала “Программирование” с просьбой открыть в журнале рубрику, посвященную обсуждению терминологических проблем.

В качестве специального тематического доклада Рабочей группе был представлен доклад Ю. В. Капитоновой, А. А. Летичевского и Н. М. Мищенко (ИК АН УССР), посвященный изложению методов реализации языка МАЯК, предназначенного для таких перспективных вычислительных средств, как макроконвейерная вычислительная система.

На заседании был утвержден план дальнейшей работы РГ, включающий проведение в 1984 г. двух заседаний:

- во Львове в апреле–мае с основным вопросом “Системное окружение языковых процессоров” и тематическим докладом по системе ДЕФИПС;
- в Новосибирске в октябре с основным вопросом “Методы реализации АТД в языках спецификаций”.

На заседании отсутствовали члены РГ: Ю. М. Баяковский, А. О. Вооглайд, К. С. Кузьмин, В. В. Луцикович, Д. И. Федосеев, М. Г. Цуладзе, В. В. Чернигов, Г. Д. Чинин, Е. Л. Ющенко.

В качестве наблюдателей на заседании присутствовали: Л. П. Бабенко, В. А. Евстигнеев, Е. Н. Капустина, А. Г. Красовский, А. А. Кубанский, А. С. Марголин, Н. М. Мищенко, Ю. В. Пастухов, С. Б. Покровский, М. Г. Рябовейтра, В. С. Сафонов, М. И. Селюн, Г. С. Цейтин, А. Л. Шмундак.

Отчет о работе Совещания разработчиков трансляторов и пользователей языка Симула

21–22 июня в Новосибирске состоялось Совещание разработчиков трансляторов и пользователей языка Симула. Совещание проводилось Рабочей группой по реализации языков программирования при Комис-

сии ПО СМО КВТ АН СССР совместно с ВЦ СО АН СССР и НФ ИТМ и ВТ АН СССР. В работе Совещания приняли участие представители из организаций страны из гг. Москва, Новосибирск, Ростов-на-Дону, Иркутск.

Пленарные доклады были посвящены следующим вопросам.

1. Развитие языка Симула и деятельность международной организации Ассоциация пользователей Симула
Докладчик — В. В. Ожольнишников, НФ ИТМ и ВТ.
2. Транслятор с языка Симула для МК Эльбрус
Докладчик — Х. Д. Дженибалаев, ВЦ РГУ.
3. Система программирования Симула ИПМ для ЭВМ БЭСМ-6 и ЕС ЭВМ
Докладчик — С. П. Бычков, МИФИ.
4. Система программирования Симула ГДР
Докладчик — В. В. Ожольнишников, НФ ИТМ и ВТ.

В выступлениях пользователей и в прениях обсуждались конкретные факты использования языка Симула, были высказаны замечания и пожелания разработчикам.

Для дальнейшей популяризации и распространения языка Симула совещание считает необходимым:

- 1) подготовить учебник по языку Симула, в котором подчеркнуть и другие (а не только моделирование) возможности Симулы;
- 2) подготовить перевод последнего описания языка Симула;
- 3) организовать аналогичное совещание или школу-семинар по языку Симула в 1984 г.

Решение семинара по языкам программирования для микроЭВМ и методам их реализации

Новосибирск, 28–30 июня 1983 г.

В семинаре приняли участие представители советских организаций, разрабатывающих программное обеспечение (ПО) средств микропроцессорной техники (МПТ). Всего в семинаре участвовали более 30 представителей из организаций Москвы, Ленинграда, Таллинна, Киева, Новосибирска, Минска, Вильнюса, Каунаса, Баку, Кишинева, Иркутска, Свердловска и других городов.

На семинаре были заслушаны доклады:

А. Д. Мильнев (Киев) «Языки для написания программ и микропрограмм в инструментальном комплексе АРМ2-05 и их реализа-

ция”.

- Ю. Н. Баранов (Ленинград)* “Язык системного программирования АОЛ/М и его реализация для микропроцессоров серии И-5580”.
- С. Н. Баранов, В. А. Кирилин, Н. Р. Ноздрунов, С. М. Селенжи (Ленинград)* “Совмещенные трансляторы и подходы к их реализации”.
- Ф. А. Каганов, Ю. В. Загубин (Москва)* “Языки программирования комплексов программ (КП), работающих в реальном масштабе времени”.
- А. С. Марков (Москва)* “Уровень языка и производительность труда программиста”.
- З. К. Карчрускас, Р. В. Ваитас, В. Ю. Валайтис (Каунас)* “Система генерации ассемблеров для программирования микропроцессоров”.
- Г. И. Погосянц (Москва)* “Средства подготовки микропрограмм в системе ФОРЭС”.
- А. В. Марин (Таллинн)* “Языки микропрограммирования для микропроцессорной техники”.
- Н. Б. Морозов, В. П. Котляров (Ленинград)* “Технологический комплекс разработки программного обеспечения с генерацией инструментальных систем для проблемных языков микроЭВМ”.
- В. П. Котляров, А. В. Самочадин (Ленинград)* “Адаптация операционной среды, средств тестирования и отладки для резидентной системы разработки программного обеспечения на проблемном языке микроЭВМ”.
- А. Г. Астановский, В. Н. Ломунов (Таллинн)* “Разработка процессора, интерпретирующего программы на языке ФОРТ”.
- Г. В. Златкис (Вильнюс)* “Диалоговая система структурного программирования”.
- Н. Ф. Фоминых (Ленинград)* “Разработка архитектуры микроЭВМ, ориентированная на статические алгоритмические языки высокого уровня (АЯВУ)”.
- А. Н. Терехов (Ленинград)* “Создание виртуальной 16-битовой машины, ориентированной на статические АЯВУ, на однобитовом процессоре”.
- А. А. Берс, Ю. В. Коваленин, М. В. Овчаренко, В. Г. Поляков, С. Б. Руднев, В. А. Четвергин (Новосибирск)* “Средства программирования для микропроцессорных систем на основе языков вместо ассемблера”.

А. А. Берс, В. Г. Поляков, С. Б. Руднев (Новосибирск) “Система средств программирования вместо универсального языка (принципы системы ЯХОНТ)”.

В рамках семинара были проведены три дискуссии.

1. Требования к языкам общения с микроЭВМ со стороны проблемных областей
(*Руководитель — В. П. Котляров*).
2. Программно-аппаратная архитектура микропроцессорных комплексов и средства обеспечения мобильности программ
(*Руководитель — А. А. Берс*).
3. Общая дискуссия по проблемам языков программирования для микроЭВМ и методы их реализации
(*Руководитель — А. П. Ершов*).

На дискуссиях выступили около 20 участников семинара.

Одним из основных вопросов докладов и дискуссий был вопрос уровня языка. Отмечалось, что разработка больших и сложноорганизованных программ на языках низкого уровня увеличивает возможность появления ошибок в программах и затрудняет их обнаружение и исправление. Это приводит к затяжке сроков разработки, снижению надежности программного продукта и затрудняет его сопровождение.

С другой стороны, кросс-системы для языков высокого уровня, как правило, сами разрабатываются медленно, располагают недостаточным ассортиментом средств отладки программ как на уровне входного языка, так и на уровне языка макропроцессора. Вырабатываемые с их помощью рабочие программы зачастую не удовлетворяют требованиям ряда применений по качеству (объем программ и/или скорость работы). В резидентных системах также затруднительно получить высокую эффективность результата при использовании языков высокого уровня. Не всегда удается учесть требования работы в реальном времени. Языки высокого уровня пока слабо ориентируются на встроенные вычисления.

Было отмечено, что одним из наиболее существенных требований со стороны проблемных областей использования средств микропроцессорной техники является требование многофункциональности языка программирования. Язык программирования должен включать в себя как средства формулировки задания и создания алгоритма, так и средства отражения пути разработки программного обеспечения, взаимодействия с имеющимися программами и данными, формирования архи-

ва задачи с историей процесса решения и т.п. Отмечался перспективный подход, продемонстрированный в системах ЯХОНТ (ВЦ СО АН СССР) и КОМПАС (ЛПИ), к средствам пользовательского окружения, ориентированный на широкое использование механизма свертки понятий.

Отмечались определенные положительные черты языка ФОРТ, предлагающего достаточно простой подход для реализации пользовательской свертки понятий.

Семинар отметил, с одной стороны, необходимость широкого внедрения существующих АРМ разработчика программного обеспечения типа АРМ2-05 с языками ассемблерного типа, кросс-систем, настраиваемых на устройства МПТ, и, с другой стороны, перспективность и необходимость расширения области применения языков типа ЯВА с заменой ими фирменных ассемблеров. Ввиду важности фактора надежности ПО для средств МПТ отмечались хорошие свойства языков со статическим контролем типов. Как для языка, так и для методов его реализации очень важным является учет специфики классов пользователей и типов микроЭВМ. Все вопросы эффективности, надежности, выбора языка должны решаться конкретно с указанием области применимости этих решений по отношению к выделенным классу пользователей и типу МПТ.

Семинар отметил большую перспективность приспособления аппаратуры к способу пользования, важность схемной реализации языков высокого уровня или их отдельных черт, учета в архитектуре МПТ языка высокого уровня и технологии его использования. Отмечалась высокая эффективность подхода к построению языковых процессоров на базе идеологии виртуальных машин, высокая скорость создания программных языковых процессоров на их основе и приемлемая для большинства встроенных применений эффективность реализации.

Важную роль для реализации программного обеспечения встроенных микроЭВМ играет применение смешанных вычислений. Отмечалось, что смешанный вычислитель позволяет в процессе конкретизации снять неэффективность проблемных программ, вносимую средствами динамического связывания, вложенными виртуальными машинами, носительской адресацией и т.д.

Было отмечено, что реализация смешанных вычислений при конкретизации программ реального времени обеспечивает корректность преобразованных программ.

Семинар отмечает недостаточность отладки ПО управляющих и встроенных микроЭВМ на кросс-системах, в связи с чем возникает необ-

ходимость дополнения их резидентными комплексами разработки программ на микроЭВМ.

В целом семинар отмечает большую эффективность, практическую и методологическую пользу проведенного мероприятия и считает целесообразным проведение подобных семинаров и в дальнейшем.

Семинар рекомендует:

- 1) провести анализ классов пользователей и типов МПТ для определения подходящих языков программирования;
- 2) усилить работу по анализу и отбору тех черт языков высокого уровня, которые обеспечивают эффективную реализацию;
- 3) усилить работу по анализу возможностей схемной реализации языков высокого уровня;
- 4) провести работу по включению в языки высокого уровня средств, ориентированных на встроенные вычисления;
- 5) внедрить в промышленное использование реализацию процессоров, основанных на идеологии виртуальных машин;
- 6) внедрить в промышленное использование подход к реализации программного обеспечения на основе смешанных вычислений;
- 7) внедрить в промышленное использование систему программирования для микропроцессоров ЯВА, учитывая ее возможности к адаптации к широкому кругу микропроцессоров и применений;
- 8) опубликовать доклады, представленные семинару и отобранные для публикации программным комитетом, поручив подготовку и редактирование сборника трудов семинара В. П. Котлярову и Р. К. Рауду;
- 9) организовать разработку типовых требований к кросс-системам для языков высокого уровня, поручив руководству РГ РЯП и ЦПГ МПТ рассмотреть возможность создания рабочей группы для такой разработки;
- 10) усилить связь и координацию работ и проводимых мероприятий по языкам программирования для микроЭВМ и методам их реализации между комиссией СМО ККВТ АН, научно-технической комиссией по языкам и системам программирования ГКНТ и РГ ТП ГКНТ;
- 11) регулярно проводить семинары по языкам программирования для микроЭВМ и методам их реализации, поручив руководству РГ РЯП и ЦПГ МПТ организацию этих семинаров и уточнение их тематики.

Отчет о деятельности за 1983 г. Рабочей группы по реализации языков программирования,

действующей при Комиссии по системному математическому обеспечению координационного комитета по вычислительной технике АН СССР.

В 1983 г. проведены следующие мероприятия.

1. Проведено заседание Рабочей группы в г. Кишиневе — 41 чел.
2. Проведены совместные семинары с Целевой подгруппой по технологии программирования микропроцессорной техники РГ ТП по языкам и методам их реализации для микроЭВМ в г. Новосибирске — около 50 чел.
3. Проведен рабочий семинар реализаторов языка Симула-67 в г. Новосибирске — около 20 чел.
4. Продолжена активная работа ЦПГ по измерению систем программирования.
5. Создана ЦПГ по анализу и классификации средств расширения в языках программирования в составе: Д. Н. Тодорой (председатель), Н. М. Мищенко, М. И. Селюн.
6. Создана ЦПГ по исследованию характеристик существующих макрогенераторов в составе: Л. Ф. Белоус, М. С. Маргалин, А. С. Марков, И. И. Пилецкий, В. Л. Темов.
7. Создана ЦПГ по проблемам терминологии в области языков программирования и методов трансляции в составе: Д. Б. Подшивалов (председатель), В. М. Курочкин, С. С. Лавров, А. С. Марков, С. Б. Покровский.
8. Члены группы приняли активное участие в проведении конференции в г. Таллинне и в г. Кишиневе и в проведении ростовской школы-семинара по смешанным вычислениям.

Информация о проведенных заседаниях и семинарах

1. С 3 по 5 июня в г. Кишиневе проходило очередное заседание группы с основным вопросом: расширяемые языки и системы.

Было заслушано 14 докладов, в которых обсуждались как общие проблемы расширяемых языков и систем, так и опыт конкретной реализации таких систем. В качестве тематических докладов заслушаны: “О терминологии в языках и методы трансляции” и “Реализация языка МАЯК”.

На заседании были утверждены три новые ЦПГ и выработан план дальнейшей работы группы, включающий проведение

в 1984 г. двух заседаний:

- во Львове в мае с основным вопросом “Системное окружение языковых процессоров” и тематическим докладом по системе ДЕФИПС;
- в г. Новосибирске в ноябре с основным вопросом “Методы реализации АД и языков верификаций”.

2. С 28 по 30 июня в г. Новосибирске проходил семинар по языкам программирования для микроЭВМ и методам их реализации.

Было заслушано 16 докладов и проведено 3 дискуссии.

Семинар рекомендовал провести анализ классов пользователей и типов микропроцессорной техники для определения подходящих языков программирования, усилить работу по отбору черт языков высокого уровня и их схемной реализации, разработать типовые требования к кросс-системам для языков высокого уровня и т.п.

Семинар отметил большую эффективность, практическую и методологическую пользу и целесообразность проведения подобных семинаров в дальнейшем.

3. С 21 по 22 июня в г. Новосибирске проходил рабочий семинар реализаторов языка Симула-67, на котором рассматривались вопросы развития языка и три конкретных реализации. Решено подготовить учебник по языку Симула-67, перевод последнего описания и организовать аналогичный семинар в 1984 г.

План мероприятий на 1984 год

Рабочей группы по реализации языков программирования

1. Провести заседание РГ РЯП в г. Львове с 21 по 26 мая, на котором рассмотреть следующие вопросы:
 - системное окружение языковых процессоров (отв. В. М. Пентковский);
 - система ДЕФИПС (отв. В. Н. Редько).
2. Провести совместное с РГ по синтезу и преобразованиям программ заседание РГ РЯП с 1 по 22 ноября с основным вопросом “Методы реализации АД и языков спецификации” (отв. Э. Х. Тыугу).
3. Вести активную работу в созданных ЦПП.
4. Рекомендовать членам РГ РЯП принимать активное участие и оказывать содействие в проведении конференций по языкам программирования и методам трансляции.

Протокол 6-го заседания Рабочей группы по реализации языков программирования

С 21 по 26 мая 1984 г. в Львовском госуниверситете проходило 6-е заседание Рабочей группы по реализации языков программирования. На заседании были сделаны 14 докладов, проведены 2 дискуссии, заслушаны краткие отчеты Целевых подгрупп РГ РЯП, продемонстрирована работа систем СУПЕР и ДЕФИПС. Основным вопросом заседания являлось системное окружение языковых процессоров.

В. М. Пентковский (Москва) в докладе “Требования к среде программирования на языке АДА” сделал обзор требований к системе программирования на языке АДА.

- Система должна иметь архив и инструментарий. При работе с архивом должна иметься возможность заведения и хранения объектов различных классов (тексты программ, программы в кодах, тестовые программы, сообщения об ошибках и т.д.) с возможностью установления взаимосвязей объектов и истории их развития. Инструментарий должен предоставлять возможность работы в терминах структур объектов архива, поддерживать модульность разработки программ, обеспечивать символьную отладку программ и т.д.
- Система должна предоставлять возможность управления версиями создаваемого программного обеспечения.

Отмечалось, что обеспечивая возможность культурного программирования, требования не затрагивают ряд вопросов, например, использование фрагментов программ, написанных не на языке АДА, проблемы надежности, работу в сети ЭВМ.

В докладе М. В. Паремского (Москва) “Система UNIX” отмечалось, что основу системы UNIX составляют файловая система и интерпретатор SHELL. Система обладает средствами форматирования текстов, автоматической трансляцией измененных файлов и др. Каждый пользователь может создать свой интерпретатор языка управления заданиями. При этом он может пользоваться средствами ядра системы (порождение и уничтожение процессов, опрос, передача сообщений, ввод/вывод и т.д.). Много внимания было уделено языку “С” и степени переносимости программ, написанных на языке “С”.

Отмечалось, что в 7-й версии UNIX переносимый транслятор с языка “С” (с выделенными местами машинной зависимости), работающий на PDP-11-45, не может работать на CM-4 из-за недостаточности ресур-

сов. Система UNIX вызвала большой интерес у участников заседания, поэтому вечером М. В. Паремский ответил на многочисленные вопросы.

А. А. Рейтсакас (Таллинн) в докладе “Система InterLisp” рассказал, что система хранит историю взаимодействия с пользователем, нумеруя его запросы. Номера запросов пользователь в дальнейшем может использовать. Взаимодействие происходит через систему окон экрана дисплея.

В докладе В. Ю. Волконского и В. М. Пентковского (Москва) “Системная поддержка СП в МВК ‘Эльбрус’” В. Ю. Волконский сообщил о динамической и технологической поддержке систем программирования. Отмечалось, что многие функции динамической поддержки реализованы аппаратно, что обеспечивает эффективное исполнение программ. Использование стандартных файлов обеспечивает скорость работы компиляторов. Богатый системный инструментарий применим ко всем языкам высокого уровня. В МВК реализовано многоязыковое взаимодействие, широко используются стандартные пакеты и модули.

В докладе А. Л. Шмундака и А. Л. Томберга (Таллинн) “Система Smalltalk-80” А. Л. Шмундак рассказал о входном языке системы, экранном текстовом редакторе, работающем с помощью “колобка”, возможности перенесения системы путем создания интерпретатора виртуальной машины.

Г. И. Сердюк (Новосибирск) в своем сообщении “Системное окружение СП и технологических комплексов БЭСМ-6” упоминал системы КРАБ, Димон, Пульт, ДМС, Дубна, Темп, Интеграл, остановившись несколько подробнее на системе Дубна.

В докладе А. Н. Терехова и А. П. Рухлина (Ленинград) “Динамическое окружение Алгола 68” А. П. Рухлин сообщил о введении во входной язык понятия модуля, об изменении направления роста динамического стека, что позволило существенно сократить затраты на вызов процедур. Из инструментария разработчикам других трансляторов могут быть предложены модули генерации объектного кода и распределения памяти. В настоящее время транслятор разрабатывается для 9 различных машин.

Л. М. Романовская (Минск) в докладе “Об операционном окружении СП Кобол в ЕС ЭВМ” рассказала о возможности использования диалогового символьного отладчика, возможности получения аварийной выдачи. Как основной недостаток ОС ЕС ЭВМ было отмечено отсутствие общей концепции.

В докладе Д. Т. Федорова и И. И. Пилецкого (Минск) “Об операционном окружении СП ПЛ/1 в ЕС ЭВМ” Д.Т.Федоров сообщил о наличии обычного, оптимизирующего и отладочного трансляторов с языка ПЛ/1 в ЕС ЭВМ с отслеживанием единства входных языков. Отмечалась возможность символьного вывода и организации собственного страничного обмена и аналога виртуальной памяти.

С. М. Шелестов (Новосибирск) в докладе “Динамическая поддержка ПЛ/1 в МВК Эльбрус” рассказал о различных видах прерываний и их обработке, о реализации эпилогов блоков через прерывания. В отладочном режиме задание на отладку транслятор оформляет отдельным модулем, а полученную при этом программу можно использовать как обычную.

В докладе “Система Кант” М. Р. Шура-Бура (Москва) остановился на вопросах тотальной автоматизации программирования. Был описан язык Кант, алфавит которого состоит из конечного числа терминальных символов и бесконечного числа терминов. Фрагменты программ, написанных на языке Кант, представляют собой последовательность терминалов и терминов, которые могут быть специфицированы. Развитие программы состоит в уточнении терминов. К классическим операторам структурного программирования добавлены завершители последовательности операторов и операторов цикла.

Е. А. Жоголев (Москва) выступил с докладом “Модульная многоязычная СП”.

Доклад В. Н. Редько (Киев) “Система ДЕФИПС” являлся докладом, предваряющим демонстрацию работы системы. ДЕФИПС предназначен для разработки интерпретаторов и трансляторов. Система состоит из ядра и фондов. В ядро системы входят препроцессоры из человеческой формы в машинную, конструкторы и сборщик. Фондами является специализированная база данных. Система в основном написана на ассемблере ЕС ЭВМ и частично на ПЛ/1. Основная “изюминка” системы — вскрытие сущности и семантики при игнорировании синтаксиса. В. Н. Редько привел классификацию функций на ординарные, структурные и интерфейсные. Аппарат аналогичен аппарату грамматик, но здесь порождается пара — смысл, синтаксис. На демонстрации представлялись диалоговый монитор ВЕКТОР, Паскалеподобный процессор, ЛИСПподобный процессор.

В. Л. Темов (Ленинград) в докладе “Системы Масон и Скоропись — открытая операционная среда” сообщил, что системы Масон и Скоро-

пись, реализованные на ЕС ЭВМ и Искра-226 соответственно, позволяют разрабатывать легко познаваемые и модифицируемые программы. Система Мэсон, не выдерживая конкуренции с промышленными трансляторами ЕС ЭВМ, позволяет тем не менее решать более сложные задачи, которые не могут быть решены с помощью штатного матобеспечения. На машине Искра-226 до разработки системы Скоропись были только трансляторы с языка БЕЙСИК и ассемблера. Единственным типом данных при работе с системами являются таблицы. Программы являются элементами таблиц, поэтому над ними можно выполнять различные операции.

Перед демонстрацией системы СУПЕР вводное сообщение сделал В. А. Серебряков (Москва). Система СУПЕР, реализованная на БЭСМ-6, ЕС ЭВМ, СМ-4 и NORD, ориентирована на разработку однопроходных трансляторов; инструментальным языком является язык Паскаль, метаязыком системы — специальный Паскалеподобный язык. Была

предъявлена документация на систему и продемонстрированы 4 тестовых примера.

В состоявшейся в конце заседания общей дискуссии выступили А. Н. Терехов, М. Е. Неменман, В. П. Котляров, Д. Н. Тодорой, В. М. Пентковский, М. П. Бабенко, В. Н. Касьянов, С. М. Абрамович, И. В. Поттосин.

Отмечалось некоторое несоответствие докладов теме заседания Рабочей группы, различное понимание понятия “окружение”. Необходимо рассматривать интегрированные системы программирования, где редакторы, документаторы, трансляторы и т.д. связаны между собой и имеют единые языки управления заданиями. Причем желательно, чтобы система поддерживала все этапы решения задачи — от постановки задачи до получения программного продукта.

Для мини-машин отмечалась необходимость создания не только кросс-систем, но и портативных комплексов с очень простыми языками, а сервисом, аналогичным сервису комплексов на больших ЭВМ (пакеты, ситуации, символьная отладка и т.д.).

Нужны пользователю и специализированные системы, основанные на базе знаний предметной области.

Говорилось о роли языка в современной системе программирования — интегральное связывание всех компонент системы. Были поставлены некоторые вопросы, связанные с разработкой интегральных

систем программирования. Например, проблемы задания семантики и прагматики систем пакетов, баз данных, используемых разрабатываемой системой, проблемы задания семантики модулей области пользователя и т.д.

Предмет обсуждения показался членам Рабочей группы настолько важным и интересным, что было решено посвятить ему заседание РГ РЯП в апреле 1985 г.

Были заслушаны краткие сообщения председателей Целевых подгрупп (ЦПГ).

Д. Н. Тодорой сообщил, что подготовлен сборник прикладной информатики с докладами предыдущего заседания РГ РЯП (июнь 1983 г., Кишинев). Подготовлены текст трех выступлений и список литературных источников по вопросам средств расширения в языках программирования.

ЦПГ, от имени которой выступал Л. Ф. Белоус, занимается анализом макропроцессора ЕС ЭВМ, универсальных макропроцессоров, а также макросредств на базе языка Рефал на ЕС ЭВМ, универсальных макропроцессоров, а также макросредств на базе языка Рефал на ЕС и СМ-4.

Заседание РГ РЯП решило включить в состав ЦПГ Ю.В. Пастухова (Харьков), а председателем избрать Л. Ф. Белоуса.

Т. С. Васючкова сообщила, что ЦПГ подготовила отчет о своей работе, которая велась как при встречах членов ЦПГ, так и путем активной переписки. Определена система понятий в виде метрики аттестации и методы оценки характеристик. Необходимость проведения экспериментов с метрикой трансляторов, содержащей около 200 свойств, определяет большую сложность работы и дает возможность ЦПГ ставить вопрос о включении данной тематики в планы работ. Неформальная подгруппа, состоящая из А. Н. Бирюкова, М. Б. Меристе и В. А. Серебрякова, занимается методами формализации описания языков программирования. Ею рассматривались системы СУПЕР, ДЕФИПС, СПТ, разработанная в Тарту на основе языка ФОРТ. Проводилась классификация атрибутивных грамматик и проверка корректности. Рассматривались методы генерации кодов.

Д. Б. Подшивалов рассказал, что работа ЦПГ носит просветительский характер. Ближайшей задачей ЦПГ является комментирование терминологического словаря операционной системы UNIX. Желательно также составить циркулярное письмо с терминологией утвержденного

стандарта на язык Паскаль для рассылки во все издательства с просьбой придерживаться указанной терминологии. Решено также послать письмо в редакцию по поводу вышедшей книги по языку Алгол 68.

Ближайшие планы работы РГ РЯП.

1. Проведение заседания РГ РЯП в г. Новосибирске 1–2 ноября 1984 г. с тематикой “Методы реализации абстрактных типов данных и языков спецификаций”.
2. Продолжение активной работы всех ЦПП.
3. Экспертирование трансляторов Алгол—Эльбрус (НФ ИТМ и ВТ).

План работы на 1985 год

1. Проведение заседания РГ РЯП в г. Баку в первой половине апреля с основной тематикой “Интегрированные системы разработки программ (обзор советских и зарубежных разработок)”.
2. Тематические доклады по методам генерации кодов (А. Н. Бирюков).
3. Проведение совещания по языкам программирования АДА (В. М. Курочкин).
4. Проведение совещания по расширяющимся системам (Д. Н. Тодорой).
5. Возможное проведение совещания по системам отладки (В. Л. Катков).
6. Продолжение активной работы всех ЦПП.
7. Участие членов Рабочей группы в организации и проведении Всесоюзных конференций по программированию.

На заседании отсутствовали следующие члены Рабочей группы: Б. А. Бабаян, Ю. М. Рябовейтра, М. Г. Гонца, К. С. Кузьмин, С. С. Лавров, А. А. Летичевский, В. В. Луцикович, Э. Х. Тыгу, В. И. Цагельский, М. Г. Цуладзе, В. В. Чернюгов, Г. Д. Чинин.

В качестве наблюдателей присутствовали около 50 человек — больше, чем на любом предыдущем заседании РГ РЯП: Д. Я. Авербух, А. Л. Александров, Т. М. Алиев, Л. М. Бабенко, С. Н. Берестовая, А. Н. Бирюков, Я. П. Бузько, А. Ю. Бяков, С. В. Веретенников, В. Н. Власенко, В. Ю. Волконский, Т. А. Гринченко, А. Д. Дагальян, В. А. Евстигнеев, Е. А. Жоголев, В. П. Жук, Б. Н. Кальняк, Р. А. Карасева, В. Л. Катков, Б. И. Коган, Г. П. Кожевникова, Я. С. Костив, А. Н. Костовский, В. П. Котляров, Л. В. Медведева, В. М. Олейник, М. В. Паремский, С. Б. Покровский, О. М. Рау, Д. А. Вейтсакас, И. Я. Ривак, Л. М. Романовская, А. П. Рухлин, Ю. А. Рыбалко, Г. И. Сердюк, Л. И. Се-

ребрянникова, А. Ф. Урахчин, А. Т. Федоров, В. Н. Чмутова, В. В. Чубук, В. С. Шевяков, С. М. Шелестов, Н. В. Шкут, А. Л. Шмундак, М. Р. Шура-Бура.

Отчет о деятельности за 1984 г.

Рабочей группы по реализации языков программирования,

действующей при Комиссии по системному математическому обеспечению координационного комитета по вычислительной технике АН СССР

В 1984 году проведены следующие мероприятия:

1. Состоялось заседание Рабочей группы в г. Львове — 67 человек.
2. Состоялось совместное заседание Рабочих групп по синтезу и преобразованию программ и по реализации языков программирования в г. Новосибирске — 51 человек.
3. Состоялось совещания ЦПГ по измерению систем программирования и по исследованию характеристик макрогенераторов.
4. По заданию ГКНТ ЦПГ по измерению систем программирования провела анкетирование большинства существующих в СССР трансляторов.
5. Члены группы приняли активное участие в подготовке и проведении всесоюзных конференций, семинаров и школ по системному программированию.

Информация о проведенных заседаниях

1. С 21 по 26 мая в г. Львове проходило 6-е заседание Рабочей группы с основным вопросом: “Системное окружение языковых процессоров”.

Было заслушано 14 докладов, проведены 2 дискуссии, заслушаны краткие отчеты Целевых подгрупп РГ РЯП, участникам заседания была продемонстрирована работа систем СУПЕР и ДЕФИПС.

Отмечалась необходимость рассматривать интегрированные системы программирования, где все компоненты связаны между собой и имеют единые языки управления заданиями.

Системы должны поддерживать все этапы решения задачи — от постановки задачи до получения программного продукта. Этот вопрос показался членам группы важным и интересным, так что было решено посвятить ему заседание РГ РЯП в 1985 году.

2. С 1 по 2 ноября в г. Новосибирске проходило 7-е заседание РГ РЯП, проводившееся совместно с Рабочей группой по синтезу и преобразованиям программ. Основным вопросом были методы реализации абстрактных типов данных и языков спецификаций.

Было сделано 7 докладов, заслушаны краткие отчеты Целевых подгрупп РГ РЯП.

Выработан план дальнейшей работы, включающий проведение в 1985 году следующих заседаний:

- заседание РГ РЯП с основным вопросом: “Интегрированные системы программирования”;
- заседание ЦПП по измерениям систем программирования для согласования некоторого варианта метрики аттестации трансляторов;
- заседание ЦПП по анализу и классификации средств расширения в языках программирования по вопросам расширяющихся языков и систем;
- тематическое совещание РГ РЯП по вопросам генерации кодов.

Руководство группы с запозданием сообщает о включении в РГ РЯП Бирюкова Александра Николаевича (Москва) в качестве наблюдателя, а также о включении Когана Бориса Исааковича (Владивосток) и Пастухова Юрия Владимировича (Харьков) в состав ЦПП по измерению систем программирования и по исследованию характеристик существующих макрогенераторов соответственно.

План мероприятий на 1986 год

Рабочей группы по реализации языков программирования

1. Провести заседание РГ РЯП по теме “Текущие проблемы разработки трансляторов”, на котором особое внимание уделить ведущимся в настоящее время в СССР реализациям языка АДА. Для этого на заседание будут приглашены представители коллективов разработчиков трансляторов.
2. Провести школу-семинар по расширяющимся языкам и системам (май 1986, г. Кишинев, проводит ЦПП по расширяющимся языкам и системам).
3. ЦПП по методам измерения систем программирования провести анкетирование пользователей по следующим вопросам:
 - какие языки программирования используются и почему;
 - какие языки программирования хотелось бы использовать и почему не используются;
 - каковы недостатки используемых трансляторов.
4. Провести собрание вновь созданной ЦПП по автоматизации построения трансляторов.
5. Вести активную работу во всех ЦПП.

6. Рекомендовать членам РГ РЯП принимать активное участие и оказывать содействие в проведении конференции по языкам программирования и методам трансляции.

Информационное письмо

Глубокоуважаемый

Извещаю Вас, что с 3 по 5 июня 1986 г. в г. Кишиневе Кишиневский госуниверситет проводит совещание-семинар Целевой подгруппы по расширяемым средствам программирования.

Список вопросов для обсуждения, подготовленный председателем ЦПГ Д. Н. Тодороем, прилагается. Если Вы или Ваши коллеги заинтересованы в участии в этом совещании-семинаре, прошу до 1 марта выслать заявку на участие и/или тезисы доклада или выступления (до 2-х страниц) по адресу: 277003, г. Кишинев, ул. Садовая, 60, ИЦ КГУ, Тодорою Дмитрию Николаевичу.

Тексты тезисов печататься не будут, они служат основой для выработки программы совещания-семинара.

Членов РГ и членов ЦПГ прошу помочь в подготовке совещания-семинара привлечением заинтересованных в тематике специалистов и инициированием подготовки докладов и сообщений.

Телефон для справок: (8-042-0) 47-07-02 (Тодорой Дмитрий Николаевич).

Председатель РГ РЯП
И. В. Поттосин

Предварительный список вопросов для обсуждения

1. Что такое расширитель в языке?
2. Что такое расширитель в системе?
3. Что такое расширяемый язык, транслятор, система, технология программирования, расширяемый продукт программирования?
4. Методы объявления, закрепления, модификации и удаления расширения в языке и системе программирования.
5. Расширяемые языки и методы их реализации.
6. Расширители в языках программирования, прагматика, синтаксис, семантика, контексты использования, примеры — разделы объявителя расширения.
7. Вертикальные, горизонтальные и нисходящие расширяемые языки и системы.

8. Расширяемые и вероятностные системы, системы познания.
9. Расширяемые и саморасширяемые автоматы, машины.
10. Базы расширений, базы данных, базы знаний.
11. Функциональное, логическое, концептуальное, вероятностное (основанное на вероятностных словарях), расширяемое программирование.
12. Расширяемые машины пятого поколения.
13. Расширяемые процессоры машинной графики.

Участники совещания-семинара могут пополнить список обсуждаемых вопросов, в том числе присылкой соответствующих тезисов.

СОДЕРЖАНИЕ

Предисловие редактора.....	3
Поттосин И. В. А. П. Ершов — пионер и лидер отечественного программирования.....	7
<i>Любимский Э. З.</i> , Поттосин И. В. , <i>Шура-Бура М. Р.</i> От программирующих программ к системам программирования (российский опыт).....	17
Поттосин И. В. А. П. Ершов и становление новосибирской школы программирования.....	28
<i>Жоголев Е. А.</i> Как А. П. Ершов стал программистом.....	41
<i>Змиевская Л. Л.</i> Чуть воспоминаний.....	43
Поттосин И. В. История Альфа-проекта.....	46
<i>Корнева Л. А.</i> История Альфа-группы.....	56
<i>Рар А. Ф.</i> История Эпсилон.....	66
Поттосин И. В. Эпсилон-история.....	73
<i>Рар А. Ф.</i> История Сигмы.....	78
<i>Степанов Г. Г.</i> История Сигмы.....	80
<i>Берс А. А.</i> , <i>Рар А. Ф.</i> Диалог об Алголе 68.....	85
<i>Покровский С. Б.</i> Внутренний язык в Бета.....	98
<i>Городня Л. В.</i> Экспериментальные системы программирования Литл, Лисп.....	103
<i>Левин Д. Я.</i> Сетл: Реализация и применение теоретико-множественного языка программирования.....	106
<i>Замулин А. В.</i> Информационные системы и базы данных.....	114
<i>Городня Л. В.</i> Откуда берутся хорошие программисты.....	117
<i>Черемных Н. А.</i> , <i>Курляндчик Г. В.</i> Библиотека и архив академика Ершова.....	124
<i>Никольников Е. И.</i> Конструкторское бюро системного программирования — КБ СП.....	130
<i>Евстигнеев В. А.</i> Научно-организационная деятельность Комиссии по системному математическому обеспечению ККВТ АН СССР.....	134
<i>Степанов Г. Г.</i> Протоколы Рабочей группы по реализации языков программирования.....	155

**СТАНОВЛЕНИЕ НОВОСИБИРСКОЙ ШКОЛЫ
ПРОГРАММИРОВАНИЯ**
(мозаика воспоминаний)

Под редакцией

проф. Игоря Васильевича Поттосина

Рукопись поступила в редакцию 25. 12. 2000

Ответственный за выпуск Т. В. Степанова

Редактор З. В. Скок

Подписано в печать 25. 12. 2001

Формат бумаги 60 × 84 1/16

Объем 11,1 уч.-изд.л., 12,1 п.л.

Тираж 100 экз.

НФ ООО ИПО “Эмари” РИЦ, 630090, г. Новосибирск, пр. Акад. Лаврентьева, 6