

И. В. Поттосин

## Российские исследования по языкам программирования и трансляции

**Предисловие редактора.** Данная работа, практически полностью повторяющая соответствующий препринт Института систем информатики СО РАН, была передана мне автором в 1996 г. К сожалению, в то время так и не представилась возможность опубликовать столь интересный обзор, в котором охвачены направления работ ведущих исследовательских центров нашей страны. С момента освещаемой ситуации минуло 10 лет. Тем не менее, читатель наверняка найдет полезным эту навигационную "карту", дающую ключ к изучению деятельности различных людей, групп и коллективов в области программирования в настоящее время. [Р. П. Богатырев]

### Вступление

Исследования по языкам программирования и трансляции всегда составляли неотъемлемую часть всех исследований по программированию практически с самого начала возникновения этой дисциплины. Они остаются и сейчас заметной частью общих исследований, хотя появление новых областей и интересов несколько потеснили их. Было бы естественно бросить взгляд на состояние отечественных работ в этом направлении.

В 1995 году вышел обзор состояния российских работ [15]. Предлагаемый текст является несколько исправленной версией этого обзора. История появления обзора такова. Наш французский коллега Пьер Непомящи, руководитель международных связей INRIA (Франция) решил представить французской научной общественности общий обзор российских исследований по программному обеспечению, который и был подготовлен и опубликован (см. [1]). Частью этого общего обзора был и подготовленный автором обзор исследований и разработок в области языков программирования и трансляции. Другим побудительным мотивом подготовки обзора было желание подвести итог работы Всесоюзной группы по языкам и системам программирования, которой в течении долгого времени автор вместе с В. М. Курочкиным и Э. З. Любимским руководил.

Обзор готовился с активной помощью большого числа российских специалистов. В организации сбора информации значительную роль сыграли В. П. Иванников, В. А. Серебряков и А. Н. Терехов. Кроме них информацию для обзора предоставили С. Н. Баранов, С. С. Гайсарян, А. Я. Диковский, И. Б. Задыхайло, С. П. Крицкий, А. В. Манцивода, В. М. Михелев, А. Е. Недоря, О. Н. Перминов, В. О. Сафонов, В. А. Хорошевский, за что им, а также П. Непомящи я глубоко благодарен. За качество же отражения их информации в обзоре ответственность лежит целиком на мне.

Обзор готовился в 1993 г., и ситуация, в нем отраженная, соответствует началу 1994 г. За истекшее время она, конечно, изменилась, однако, автор надеется, не настолько, чтобы коренным образом изменить картину. Исследования эволюционировали, но не очень резко.

Не имея возможности полностью модернизировать обзор, автор сознательно не менял изложение и там, где определенное изменение положения и было ему известно, сохраняя точный синхронизованный характер временного среза --- на начало 1994 г. Поэтому термин "сейчас", "в настоящее время", "в текущий момент" надо связывать с началом 1994 г.

### 1. Предыстория

Традиционно в России существовало значительное число групп, активно занимавшихся исследованиями и разработками в области трансляции. Эти группы функционировали в академических институтах (Институт прикладной математики АН СССР, Вычислительный центр АН СССР, Вычислительный центр СО АН СССР, Институт точной механики и вычислительной техники АН СССР и др.), в вузах (Московский, Ленинградский, Ростовский университеты и др.), а также в ряде промышленных исследовательских институтов. Необходимость создания широкого спектра

трансляторов для разнообразной отечественной техники (а подавляющее большинство используемых средств вычислительной техники было отечественным) стимулировало как фундаментальные исследования, так и экспериментальные разработки в этой области. Несколько повлияла и определенная закрытость советского общества, которая сказывалась и в ограниченности научных связей, и в затрудненности развитой кооперации с зарубежными исследованиями, и естественном и широком вращении в международный научный поток. В результате приходилось в большей мере рассчитывать только на собственные силы, дублировать исследования, искать собственные пути, не всегда оправданные. Отсюда и другой эффект --- из-за ориентации на внутреннего читателя большинство научных публикаций выходило на русском языке, и многие интересные результаты оставались не известными международному сообществу.

В последние годы число групп и интенсивность работ заметно уменьшились. Отчасти это естественный процесс --- наличие на международном рынке большого числа приемлемых трансляторов для используемой вычислительной техники (а зарубежная техника, может быть, не в лучших ее вариантах, доступна отечественному пользователю и доминирует на отечественном рынке) снизило интерес потребителя к новым разработкам по системам программирования и сместило его к различным прикладным и проблемно-ориентированным системам. Сказались также кризис отечественной промышленности в области вычислительной техники и серьезное сокращение государственного финансирования исследований (а других источников финансирования науки в стране пока, по существу, нет).

Вращение российских исследований по трансляции в мировой процесс также идет непросто. Политические ограничения в значительной мере заменились финансовыми. Поддержание необходимых постоянных контактов, требует пусть небольших, но зачастую непосильных затрат. Сказываются и определенные последствия былой замкнутости. Даже внутрироссийские связи и контакты заметно ухудшились из-за финансовых проблем.

Вместе с тем сохранился и продолжает работу ряд групп (основные перечисляются далее).

В данном обзоре описывается текущее состояние российских исследований и разработок в привязке характеристике активных групп.

Наиболее интересные исследования и разработки ведутся по следующим направлениям:

- Теоретическое и методические исследования процессов трансляции и преобразования программ и методов построения трансляторов и других языковых процессоров. Наиболее "продвинутые" работы здесь ведутся в области применения трансформационного подхода, методов верификации трансляторов, систем построения трансляторов, методов реализации функционально-логических языков.
- Разработка новых языков и систем программирования для них. Эти работы связаны с созданием расширений существующих языков для параллельных вычислений, или с разработкой декларативных языков --- проблемно-ориентированных, языков спецификаций, функционально-логических. Существенно, что создание языков ведется "под себя", т. е. группа, создающая язык, сама реализует или предполагает реализовать его.
- Интеграция трансляции в развитые окружения программирования. Работы этого направления прежде всего связаны с созданием технологических инструментов разработки программ для встроенных или параллельных ЭВМ.

Координацию основных исследований и разработок в области трансляции вела долгое время Рабочая группа по языкам и системам программирования, существовавшая при Комиссии по системному математическому обеспечению Отделения информатики, вычислительной техники и автоматизации АН СССР, созданной и руководимой (до 1989 г.) академиком А. П. Ершовым, а затем членом-корреспондентом АН СССР В. П. Иванниковым. С распадом Союза (и Академии наук СССР) эта группа прекратила работу. В 1991 г. (фактически на смену ей) возникла российская секция SIGPLAN (Special Interest Group on Programming Languages) ACM, однако общий финансовый кризис науки и образования существенно снизил ее реальное влияние на дела и исследования.

## 2. Активные коллективы и проводимые работы

**2.1.** Одним из ведущих российских центров по работам и исследованиям в области языков и трансляции является Институт прикладной математики РАН, где был создан один из первых советских трансляторов. Существенная часть широко использовавшихся советских трансляторов — для Алгола 60, Фортрана — также разработана в этом институте. Ряд научных групп института внесли большой вклад в становление советских исследований в области языков и трансляции.

Работа этого коллектива, руководимого И. Б. Задыхайло и В. А. Крюковым, связана с созданием системного программного обеспечения для специальных ЭВМ и больших проектов, в частности, в области космоса, атомной промышленности и нефтегазоразведки. Коллективом разработаны распределенные операционные системы для специальных применений, бортовая операционная система для космического корабля "Буран" и т.п. Работы по трансляции были подчинены указанным направлениям — создан транслятор для Симулы 67, транслятор для расширения Фортрана применительно к параллельным и векторным вычислениям с векторизатором и пр.

Коллектив включает около 60 исследователей и программистов, в проектах участвуют студенты Московского университета.

В данное время в коллективе существует проект, объединяющий три направления:

- разработка операционных систем для мультипроцессорных ЭВМ с распределенной памятью;
- разработка трансляторов для ряда языков и окружения программирования для них, предназначенных для использования в мультипроцессорных ЭВМ;
- создание окружения для проектирования систем реального времени.

Работы по трансляции ведутся в контексте всего проекта. Наиболее существенная их часть включена в работы по окружению для разработки переносимых параллельных программ — системе RAMPA, предназначенной для MIMD-компьютеров с распределенной памятью.

Система RAMPA [2] должна интегрировать три различных подхода к языкам программирования для распределенных и параллельных систем, а именно:

- языки параллельного программирования для распределенных систем, основанные на коммуникации с посылкой сообщений;
- расширения последовательных языков, связанные с разделением вычислений между процессорами в виде специальных комментариев;
- декларативные языки спецификаций.

Для первых двух целей разработано два расширения Fortran 77 — Fortran GNS и Fortran DMV соответственно.

Fortran GNS использует существовавшую ранее идею абстрактной GENESIS-машины, в которой задача распадается на множество подзадач, динамически создающихся на процессорах распределенной системы. Коммуникации между ними осуществляются с помощью посылки и приема сообщений разного типа. Подзадачи имеют только локальные данные, общие данные погружаются в общую файловую систему. Абстрактная GENESIS-машина реализована как совокупность процедур (для нескольких архитектур), поэтому есть возможность задать прямо в Fortran 77-программе распределенные вычисления. Программа на Fortran GNS (дополнительно к Fortran 77) содержит описание конфигурации и операции сообщения (трех типов). Первая версия транслятора с Fortran GNS реализована на IBM PC.

Fortran DVM (Distributed Virtual Memory) основан на параллелизации путем отображения данных на распределенную память и на управлении доступом к виртуальной памяти. Ряд средств языка аналогичен средствам Fortran D и HPF (High Performance Fortran). Кроме того, язык содержит средства для балансировки загрузки процессоров (можно определять группы процессоров и учитывать их специфику) и средства для оптимизации межпроцессорной коммуникации (определение оптимальных путей, буферизации и пр.).

Применительно к вычислительным задачам математической физики и аналогичным проблемам (особенно к сеточным методам) разработан декларативный язык спецификаций Норма, в котором задача описывается совокупностью вычислительных формул и условий на индексы, т.е. в привычной для проблемного специалиста форме. Сам вычислительный алгоритм синтезируется автоматически по этому описанию, причем в процессе синтеза может быть оптимально приспособлен к конкретной архитектуре. Ранее была создана экспериментальная версия транслятора с языка Норма в последовательный Fortran 77 и векторно-параллельный Fortran VP. Сейчас ведется разработка транслятора для MIMD-машин с распределенной памятью.

Для системы RAMPA разрабатываются также отладочные и мониторинговые средства, в частности, с моделированием многопроцессорных архитектур на инструментальной ЭВМ.

В рамках общего проекта ведутся также работы по созданию транслятора с расширенного Фортрана (Extended Fortran) для российского суперкомпьютера ES-1191 (4 скалярных, векторный и управляющий процессоры) и трансляторы для Модула-2 и Си на транспьютероподобные RISC-процессоры.

Интересной особенностью значительного числа разработок, ведущихся в этом коллективе, является то, что в качестве инструмента создания транслятора используется язык Рефал.

Рефал — давний язык (к сожалению, пока слабо распространенный), предложенный В. Ф. Турчиным, — предназначен для обработки символьной информации, его концептуальной логической моделью являются нормальные алгорифмы Маркова. Рефал-программа есть совокупность фраз, каждая из которых содержит образец (типовое выражение, используемое для наложения на объектное выражение) и "правила преобразования", тем самым представляет собой своеобразное правило переписывания термов. Семантика языка фиксирует стратегию преобразования.

В институте существует небольшой коллектив (около 5 исследователей) под руководством С. А. Романенко, который занят проблемами, связанными с Рефалом. В группе разработан транслятор для Рефала, показавший свои неплохие качества для реализации трансляторов и других подобных систем. Как принято для ряда декларативных языков, транслятор переводит Рефал-программу в код абстрактной Рефал-машины с соответствующей организацией памяти (в виде двусвязных списков). Реализация Рефал носит полупромышленный характер. В группе ведутся также работы по преобразованию Рефал-программ, в частности, создан смешанный вычислитель для Рефала.

В этом же институте существует небольшая группа (руководитель В. М. Михелев), которая занята исследованием метасредств построения языковых процессоров (т.е. любых процессоров, входная информация для которых определяется грамматическими правилами), основанных на методах объектно-ориентированного программирования. Создан язык Маркиз и метапроцессор для него. Входной язык Маркиз является объектно-ориентированным языком, на котором описание языкового процессора базируется на представлении синтаксических классов входного языка процессора классами объектов с определенными на них операциями. Определение операций осуществляется так, что возможно задание многопроходных языковых процессоров. Сейчас ведутся эксперименты с системой.

**2.2.** Отдел систем программного обеспечения Вычислительного центра РАН (теперешний руководитель — В. А. Серебряков) тоже один из старейших российских коллективов, занимающихся проблемами трансляции. Здесь разрабатывались одни из первых в СССР трансляторов, в дальнейшем был создан ряд трансляторов для отечественных ЭВМ — от М-20 до БЭСМ-6. Диапазон языков тоже весьма велик — Алгол 60, ПЛ/1, Паскаль и т.д.

Некоторые из созданных трансляторов остались экспериментальными разработками, часть же широко использовалась в СССР.

Сейчас в отделе около 15 сотрудников. Кроме них в работах участвуют аспиранты и студенты Московского университета и Московского физико-технического института.

Определяющими для последних работ этой группы являются исследования и разработки по применению атрибутивных грамматик к построению трансляторов и созданию языков и систем программирования для высокопараллельных многопроцессорных архитектур.

Исследования по атрибутивным грамматикам имели своей целью создание эффективной атрибутивной техники, позволяющей строить для современных универсальных языков трансляторы промышленного качества. Были разработаны эффективные алгоритмы атрибутивных вычислений, предложена практически удобная модель атрибутивных грамматик, продуманы механизмы построения многопроходных трансляторов, исследованы способы атрибутивного определения хороших генераторов кода. Все эти исследования были положены в основу системы построения трансляторов (СПТ) Super [3].

Система Super использовалась в качестве средства построения различных трансляторов, как для проблемных входных, так и для Модуля-подобных языков. Она показала свои возможности для построения трансляторов, которые обладали достаточной скоростью трансляции и неплохой эффективностью порождаемого кода. Наиболее показательным было применение СПТ Super для Модуля-подобных языков. Система применялась также для построения двухпроходного переносимого Модуля-транслятора и транслятора для языка программирования баз данных Модуля-90 (расширение Модуля-2). Атрибутивная техника была использована для создания достаточно изоциренных генераторов кода для нескольких ЭВМ.

В последнее время группа занимается созданием языков и систем программирования для транспьютерных архитектур.

Для параллельных вычислений в научных приложениях был разработан язык SYNAPS [4], — расширение С. Как и в работах п. 2.1, язык SYNAPS отталкивается от последних исследований по расширениям Фортрана (Fortran D, HPF, Vienna Fortran). В основу модели вычислений положены процессоры с распределенной памятью. Распределение данных осуществляется двумя основными способами — либо значения копируются на все процессоры, либо значения (составные значения — векторы и матрицы) распределяются между процессорами. Вводятся достаточно простые и наглядные параллельные конструкции, связанные с выделением параллельных ветвей и параллелизуемыми циклами. Как распределение данных, так и синхронизация действий, исполняемых на разных процессорах действий привязаны к этим параллельным конструкциям. Все это способствует тому, что структура программы соответствует структуре данных, чего нет в вышеупомянутых вариантах Фортрана. Это и ряд других достоинств языка SYNAPS делает его достаточно удобным и наглядным средством для описания параллельных алгоритмов, исполняемых на MIMD-архитектурах с распределенной памятью. Кроме того, SYNAPS явно ориентирован на научные вычисления (задачи матфизики и пр.), и ряд примеров показывает его естественность для этих задач, прежде всего для параллельной обработки больших массивов и матриц. Предложенные принципы расширения применимы не только для Си, но и для других языков (например Фортрана).

Сейчас на PC и SPARCstation разрабатывается транслятор для языка SYNAPS. Основная техника создана, и первая экспериментальная версия транслятора уже проверяется.

Для отображения Оккам-программ на произвольную транспьютерную сеть разработана система DINAMO, использующая временные характеристики алгоритма для получения наилучшего отображения. Ведутся также работы по созданию библиотек для транспьютерных приложений.

В этом же институте был реализован язык Рефал/2 (о языке см. п.2.1). В силу ориентации на приложения к задачам искусственного интеллекта и представления знаний включена в эту систему программирования и соответствующая библиотека машинных операций. Создано окружение программирования для Рефала, аналогичное турбо-системам. Все это реализовано для MS DOS и VS Windows.

**2.3.** В связи с созданием системного программного обеспечения вычислительной системы "Электроника ССБИС" (советская ЭВМ с Cray-подобной архитектурой) в Институте проблем кибернетики РАН сложился коллектив под руководством В. П. Иванникова (На основе этого коллектива в 1994 г. был создан Институт системного программирования РАН.) В рамках общих работ группой под руководством С. С. Гайсаряна были реализованы оптимизирующие

компиляторы с языков Fortran 77, Pascal, ANSI C и кросс-компилятор с языка PL/1. Для обеспечения достаточно высокого качества генерируемого кода были исследованы вопросы распараллеливания последовательных программ: векторизации итеративных циклов и планирования кода для процессоров с несколькими параллельно работающими скалярными и векторными конвейерными функциональными устройствами.

В связи с тем, что работы по ВС "Электроника ССБИС" приостановлены, группа переключилась на создание разработки и реализацию мобильных систем программирования для различных классов вычислительных систем.

Разработано расширение языка ANSI C для векторных и суперскалярных вычислительных систем (язык C []). Заканчивается реализация первой версии мобильного компилятора с этого языка, настраиваемого на конкретную архитектуру. При разработке компилятора используются современные инструментальные средства, в частности, Karlsruhe Toolbox for Compiler Construction.

В настоящее время группа работает над расширением языка C [] для массивно-параллельных вычислительных систем и создает для этого расширения мобильную настраиваемую систему программирования, ориентированную на разработку масштабируемых (scalable) программ. Первую реализацию этой системы программирования предполагается осуществить для вычислительной системы Parsytec GC в среде ОС PARIX.

Продолжаются работы по методам и алгоритмам распараллеливания последовательных программ (векторизации циклов и планированию и балансировки кода). С использованием инструментальной системы SIGMA, разработанной в Великобритании и полученной из INRIA (Франция).

В рамках создания свободно распространяемого программного обеспечения (Free Software) в сотрудничестве с Free Software Foundation (США) реализован мобильный компилятор (front-end) с языка Модула 2 для системы GNU CC.

**2.4.** На факультете кибернетики Московского инженерно-физического института под руководством О. Н. Перминова существует группа по языкам и системам программирования, состоящая из 8 исследователей с привлечением более чем десятка аспирантов и студентов.

Основная тематика работы группы — реализация Ады и ее окружения, а также исследования в этом и смежных направлениях. Сейчас эта группа осталась по существу одной из двух в России, всерьез занятой реализацией Ады. Группа отслеживает развитие Ады: она связана с международным проектом Ада 9х, Перминов является членом международного комитета по этому проекту.

Исторически группа начинала с создания трансляторов для ЕС ЭВМ. Для этой архитектуры был разработан транслятор с Паскаля (достаточно широко использовавшийся) и Ады (находился в экспериментальной эксплуатации в нескольких организациях). Далее были созданы транслятор и окружение для Ады на IBM PC совместимых компьютерах. Эта система прошла определенную проверку на достоверность, однако об ее свойствах говорить трудно, так как она широко не использовалась. Это не упрек создателям, а лишь констатация того факта, что пока Ада не нашла в России широкого или даже всерьез заинтересованного пользователя. Во многом это связано, на мой взгляд, с тем, что внедрение технологии Ада-программирования в реальные большие проекты требует предварительных затрат на обучение этой технологии, в противном случае употребление Ады (по сравнению с Паскалем, Си и Модулой-2) становится неоправданным.

Текущие работы группы связаны с созданием кросс-систем программирования для Ады на Cray-подобные и транспьютерные архитектуры. Инструментальной машиной выступает IBM PC и фрагменты созданного на IBM PC транслятора используются. Группа также ведет исследования по автоматической параллелизации Ада-программ (в смысле параллелизма процессов).

**2.5.** Один из основных российских коллективов, ведущих исследования по языкам и системам программирования, сложился в Новосибирске, в Вычислительном центре СО РАН. Становление этого коллектива и его известные работы по теории трансляции и оптимизации программ, по оптимизирующим трансляторам, по многоязыковым транслирующим системам связаны с именем академика А. П. Ершова.

С 1990 г. коллектив существует как отдельный институт — Институт систем информатики (ИСИ) СО РАН — и ведет исследования по теории программирования, архитектуре высокопроизводительных ЭВМ, системному программному обеспечению и системе искусственного интеллекта. Общее число научных сотрудников — около 90, из которых примерно половина связана с теми или иными исследованиями по языкам и системам программирования.

В лаборатории теоретического программирования института (руководитель В. А. Непомнящий) ведутся работы по верификации программ и языкам спецификации. Разработан язык спецификации систем реального времени REAL 92 [5], сочетающий средства исполнимых спецификаций (базирующиеся на известном языке спецификаций систем связи SDL) и логические средства, основанные на динамической логике процессов. Начаты исследования по алгоритмам реализации языка REAL 92 и верификации свойств, выраженных на этом языке, в частности, проведены экспериментальные реализации некоторых случаев model-checking.

Работы по верификации программ основаны на идее проблемно-ориентированной спецификации. Ориентация на класс задач позволяет осуществлять процесс верификации автоматически. Такой подход требует создания специализированных доказателей, ориентированных на проблемные области. Они основаны на мощной базе знаний, состоящей из аксиом, которые описывают свойства понятий языка спецификации. Для такой базы знаний реализуется модуль расширений, позволяющий добавлять новые аксиомы в естественной нотации. Предложен новый класс аксиом, который дает возможность автоматизировать разбор случаев, возникающих при проблемно-ориентированной спецификации. Создана система верификации, ориентированная на программы сортировки массивов, линейной алгебры и трансляции.

Верификация трансляторов в этой системе заключается в доказательстве правильности программных модулей трансляторов относительно их спецификаций по методу Хоара. Для этого созданы функциональные средства спецификаций, с помощью которых можно описать этапы работы по переводу программ с одного языка на другой, а также описан метод аннотирования процедур трансляторов, позволяющий выражать требуемые свойства. Создано аксиоматическое представление, которое дает возможность однозначно определять функции и ориентировано на автоматический перевод их в программные конструкции. В подсистеме верификации трансляторов наряду с известными правилами вывода аксиоматической системы Хоара применяются новые правила для процедур и файлов. Осуществлен важный программный эксперимент: предложенные средства использованы для спецификации трансляторов с подмножеств языка Паскаль и Бейсик, проведено аннотирование этих трансляторов и доказана их правильность с помощью системы верификации.

Ряд исследований института связан с трансформационными методами конструирования программ, в том числе и самих трансляторов.

В группе смешанных вычислений под руководством М. А. Бульонкова ведутся исследования по развитию методов преобразования императивных и функциональных программ путем смешанных вычислений. Найден новый метод организации смешанных вычислений, основанный на поливариантной схеме, что открыло путь для построения автоматических самоприменимых процессоров смешанных вычислений. По отношению к трансляции это позволяет выводить и обосновывать специфические понятия, такие как таблица символов, шаблоны кодогенерации и пр., исходя из универсальных механизмов смешанных вычислений. Было показано, что смешанные вычисления в такой трактовке дают возможность выделить из интерпретатора фазы трансляции, обосновывать константные вычисления в трансляторах. Для функциональных программ первого порядка был предложен практический метод реализации поливариантного анализа периода связывания. Инструментом для экспериментов служит автопроектор Similix, разработанный в Копенгагенском университете для языка Scheme и модифицированный в данной группе.

В лаборатории оптимизации и преобразований программ (руководитель В. Н. Касьянов) ведутся теоретические исследования и разработка экспериментальных систем, связанных с созданием процессоров преобразования программ [6]. В значительной мере работы этой группы по

преобразованию программ концентрируются вокруг концепции конкретизации программ. Конкретизация программ призвана получать качественные специализированные программы из универсальных за счет того, что входной язык расширяется средствами формализованных комментариев-аннотаций, а исходная универсальная программа (преобразуемая) аннотируется пользователем, который в аннотациях задает используемую в специализирующих преобразованиях известную ему специфическую информацию о свойствах программы и конкретных (суженных) условиях ее применения. Такими условиями могут быть сужение множества допустимых входных и выходных данных, дополнительные сведения о процессе исполнения и пр.

Были исследованы проблемы корректности конкретизирующих преобразований, их классификации, типов аннотирующих утверждений. В связи с проблемами оптимизации и преобразования программ проведен сравнительный анализ форм промежуточных представлений программ, ориентированных на оптимизацию и параллельную обработку, подготовлен каталог реструктурирующих и оптимизирующих преобразований программ для различных архитектур, включая параллельные. Создан ряд экспериментальных систем специализации, основанных на редуцирующих конкретизациях.

К исследованиям по конкретизации примыкают работы лаборатории по созданию анализаторов семантических свойств. Такие анализаторы выявляют подозрительные семантические конструкции в тексте программы и сообщают пользователю о свойствах состояний вычислений в интересующих его точках, тем самым гарантируя отсутствие в отлаживаемой программе определенных классов содержательных ошибок, выявление которых традиционными методами весьма трудоемкий процесс. Заканчивается создание анализатора свойств Модуля-программ. Перечень свойств, им выявляемых, включает корректность цепочек от задания к использованию переменных, а также программные инварианты, представляющие собой множество равенств термов, истинных в любом состоянии памяти, с которым можно прийти в данную программную точку.

В лаборатории системного программирования (руководитель И. В. Поттосин) работы по трансляции ведутся в рамках более общего проекта, имеющего целью создание методики и инструментальной поддержки разработки надежных и качественных программ для бортовых ЭВМ. Работа над проектом включает как исследования и экспериментальные разработки, так и создание развитого окружения программирования для бортовых ЭВМ методами кросс-трансляции на инструментальных ЭВМ типа IBM PC (система СОКРАТ [7]).

Наличие в окружении большого набора разнообразных инструментов работы с программами — оптимизаторов, анализаторов, преобразователей, интерпретаторов, генераторов кода — стимулировало поиск новых подходов к интеграции транслятора в окружение программирования. Был разработан и реализован операционный интерфейс front-end транслятора с остальными инструментами окружения. Операционный интерфейс характеризуется специфицированной совокупностью модулей, определяющих реакцию на обнаружение front-end'ом синтаксических и семантических конструкций входной программы. Каждый инструмент обладает своей реализацией специфицированных модулей, выполняющих нужный перевод конструкций или интерпретацию ситуации.

Оптимизацию программ в системе СОКРАТ реализует глобальный оптимизатор и оптимизирующий генератор кода. Глобальный оптимизатор осуществляет потоковый анализ программ и некоторую совокупность машинно-независимых оптимизаций (чистка циклов, удаление неиспользуемых фрагментов и вычислений и т.п.). Для оптимизаций, им выполняемых, была доказана корректность. Глобальный анализ программ осуществляется не только в интересах собственно глобальной оптимизации, но и в интересах других инструментов — специализатора, оптимизирующего генератора кода. Глобальный оптимизатор существует как отдельный языковый процессор, слабо зависящий от входного языка. Оптимизирующий генератор кода осуществляет большое число архитектурно-зависимых оптимизаций. Наличие различных архитектур бортовых ЭВМ требует сравнительно легкой настройки генератора кода на архитектуру объектной ЭВМ, и сейчас ведутся исследования по способам такой настройки.

Высокие требования к надежности разработки и эффективности бортовых программ обусловили включение в систему ряда процессоров автоматической обработки программ — специализаторов, анализаторов свойств, о которых говорилось выше. Для нужд всех таких процессоров, включая глобальный оптимизатор, создан адекватный их потребностям внутренний язык, основанный на унифицированном формализованном описании семантики исполнения входных языков.

Для повышения надежности разработки программ в системе предложен стиль структурного конструирования и реализованы экспериментальные версии инструментов его поддержки. Структурное конструирование основано на построении программ путем структурного редактирования и перманентном получении сведений о статических и динамических свойствах еще незавершенных программ. Инструменты структурного конструирования включают структурный редактор, анализаторы статических свойств (наличия неиспользуемых объектов, информационных зависимостей в программе и пр.) и интерпретатора незавершенных программ, обнаруживающего свойства исполнения.

Теперешняя версия системы СОКРАТ разработана для языка Модула-2, однако большинство инструментов системы языково-независимо и ориентировано на класс языков со статическим контролем типов — Модула-2, Оберон, Ада, С++. Определенная проработка погружения других входных языков ведется: на класс таких языков ориентирован внутренний язык и ряд процессоров обработки программ, структурный редактор является языково-настраиваемым и т. п.

ИСИ СО РАН — ведущий российский центр по работам, связанным с языком Модула-2 и его преемниками. В этом коллективе был разработан первый советский транслятор для Модулы-2 (в рамках многоязыковой транслирующей системы), созданы семейства процессоров и рабочая станция КРОНОС с архитектурой, ориентированной на поддержку Модула-программ, система программирования с языка Модула-2 для этих ЭВМ. Для семейства КРОНОС Модула-2 была базовым языком программирования, все системное программное обеспечение писалось на этом языке. Коллектив поддерживает рабочие контакты с группой автора Модулы-2 проф. Н. Вирта из Цюрихского технического университета. Институт является организатором российской группы по стандартизации Модулы-2.

Последняя работа в этом отношении — создание группой А. Е. Недори [8] биязыковой системы Mithrill. Система предназначена для разработки переносимого программного обеспечения на языках Оберон-2 и Модула-2 и позволяет смешивать в одном проекте модули на разных языках, т.е. импортировать Модула-2 — модули из Оберон-модулей и наоборот. Как близость, так и взаимодополняемость входных языков (расширение типов и методов в стиле объектно-ориентированного программирования, сборка мусора в Обероне-2, низко-уровневое программирование, отсутствие динамической поддержки в Модуле-2) делает естественным объединение входных языков как средства разработки программ и позволяет заметно увеличить общий потенциал системы программирования.

Система включает в себя несколько генераторов кода (в том числе генератор кода для Intel 80386/486) и генератор текста на языке ANSI C. Она работает в нескольких OS для IBM PC, а транслятор в ANSI C — в ряде платформ, в том числе MS-DOS, VAX/VMS, Sun SparcStation, HP-9000, Silicon Graphics. Система легко переносима на другие платформы и допускает быстрое подключение других генераторов кода, так что представляет собой достаточно приемлемый программный продукт.

**2.6.** В Российском институте искусственного интеллекта, тесно сотрудничающем с ИСИ СО РАН, существует небольшая (3-4 человека) группа под руководством Д. Я. Левина, которая является одной из немногих в мире, имеющих опыт реализации языка SETL. В группе реализовано несколько трансляторов для языка SETL. Последняя реализация — транслятор и окружение для языка SETL на IBM PC в среде Turbo-Vision. Принципы реализации этой системы и предыдущих ей основаны на абстрактной SETL-машине. Трансляция идет в коды этой машины. В результате многолетних экспериментов выбрана архитектура такой машины, хорошо отражающая теоретико-множественные средства языка. Система имеет полупромышленный характер, используется в нескольких организациях для прототипирования систем искусственного интеллекта.

**2.7.** В отделе программного обеспечения высокопроизводительных ЭВМ (supercomputers) — теперешний руководитель В. Э. Малышкин — в рамках общих работ по созданию многопроцессорных и параллельных архитектур, системного программного обеспечения для них и технологии параллельного программирования вычислительных задач ведутся исследования по языкам программирования для специальных архитектур и систем программирования для них.

Основной работой в этом направлении является создание средств программирования для крупноблочных многопроцессорных систем. В подобных архитектурах вычислительная система строится как иерархия, элементами которой являются ЭВМ. В соответствии с этим подходом была разработана мультипроцессорная вычислительная система Сибирь, а для программирования на этой системе — язык ИНЯ. Язык, являясь расширением Фортрана, содержит дополнительно средства для инициации процессов, параллельно исполняемых на подпроцессорах иерархии, их синхронизации, декомпозиции данных и т.п. Система программирования для этого языка включает в себя препроцессор, переводящий ИНЯ-программы в Fortran-программы на хост-машине, и библиотеку программ, поддерживающих распределение процессоров и данных в вычислительной системе.

Для описания и моделирования алгоритмов для ассоциативных SIMD-процессоров разработан STAR — паскалеподобный язык, имеющий такие типы данных, как битовые матрицы, слайсы (slice), слова и операции, осуществляющие ассоциативную параллельную обработку этих данных. Планируется создание соответствующей системы для моделирования STAR-программ.

**2.8.** На математическом факультете Иркутского университета сложилась активная группа под руководством А. В. Манциводы, которая занимается проблемами функционального и логического программирования и состоит из 5-6 человек (аспиранты и сотрудники).

Исследования по указанным проблемам концентрируются вокруг разработанного в этой группе функционально-логического языка Flang [10]. Flang основывается на недетерминированных функциях, в которых обычные для функциональных программ определения функций расширены возможностью употреблять прологовские отношения, а вычисления функций могут быть связаны с бэктрекингом и поиском альтернативных решений. В язык также введены средства (специальные примитивы) для задания ограничений (constraints). Все эти средства языка Flang позволяют исследовать совокупность современных парадигм программирования.

Разработан Flang-транслятор, который содержит ряд интересных решений. Для языка Flang предложена машина (Flang-машина), альтернативная машине Уоррена (WAM), и отличающаяся от нее методом восстановления переменных при бэктрекинге — в ней он основан на нумерации точек возврата (time stamping). Flang-машина, позволяющая осуществить более изощренный механизм учета ограничений, оказалась очень удобной для реализации оптимизационных механизмов в расширенном логическом программировании, в частности, глубокого возврата (intelligent backtracking). Все это в сочетании с разветвленным глобальным потоковым анализом позволяет существенно оптимизировать исполнение Flang-программ. В экспериментах Flang-транслятор показал возможность строить весьма эффективные программы (сравнимые при адекватности алгоритмов с программами, полученными Турбо-Паскалем).

Сейчас разрабатывается новая версия Флэнг-машины, эффективно поддерживающая механизм демонов, крайне важный при решении логическими средствами комбинаторных задач высокой сложности.

Основной целью группы является не создание коммерческого продукта, а дальнейшие исследования, в частности по уточнению стратегии вычислений и анализу программ с ограничениями. В целом задачей поиска является нахождение как хороших языковых средств, сочетающих выразительную мощность и возможность эффективной реализации, и алгоритмов анализа и трансляции, обеспечивающих достаточно высокую эффективность исполнения программ.

Большое внимание уделяется поиску новых стратегий вычислений логических программ, развивающих метод удовлетворения ограничений на конечных областях, а также способам реализации этих методов. Вместе с тем возможно создание и коммерческой версии, ориентированной на решение задач высокой комбинаторной сложности (составление расписаний, дискретной оптимизации и т. д.)

**2.9.** Ряд работ, сопряженных с трансляцией, ведется в отделе экспертных систем Института автоматизации и процессов управления ДВО РАН (руководитель — А. С. Клещев). Все эти работы,

как правило, связаны с языками и системами искусственного интеллекта. Разработано несколько продукционных языков представления знаний, для которых созданы системы программирования. Последним является язык РЕПРО [11], основанный на декларативных системах продукций. В процессе работ исследовались проблемы: анализа информационных связей в системе продукций; оптимизации логического вывода по числу операций вывода; преобразования продукционных программ, в том числе путем смешанных вычислений; генерации паскалевских программ по декларативному описанию базы знаний на языке РЕПРО.

В группе проводились эксперименты по применению методов обработки знаний к некоторым задачам трансляции. Была разработана экспериментальная система, моделирующая создание процессоров оптимизации путем задания в терминах баз знаний оптимизирующих преобразований программ.

Ведутся также работы по методике и инструментальной поддержке измерений качества программных средств, разработаны измерительные языковые процессоры и анализаторы нескольких языков программирования (Фортрана, Паскаля, продукционных языков для баз знаний и пр.).

**2.10.** В Институте точной механики и вычислительной техники РАН (руководитель коллектива — Б. А. Бабаян) и его Новосибирском филиале (директор — Г. Д. Чинин) создавались системы программирования для известного советского проекта суперкомпьютеров Эльбрус. Для Эльбрус-1 и Эльбрус-2 был выполнен ряд интересных работ: созданы эффективный транслятор для специально ориентированного на архитектуру Эльбрус языка высокого уровня Эль-76, трансляторы с языков Пролог и Ада, оптимизирующий транслятор с Фортрана (допускавший целый ряд диалектов Фортрана) и т.п.

Следующим шагом всего проекта должно было стать создание Эльбруса-3 с VLIW-архитектурой и модульно-конвейерного процессора (МКП) с векторной архитектурой. Для этих процессоров были разработаны трансляторы с Эль-76 и Фортрана.

Первый из них, созданный в группе Бабаяна, при генерации кода на Эльбрус-3 достигал высокого качества объектного кода за счет хорошего использования конвейера и изоощренной упаковке VLIW-кода. Был разработан ряд оригинальных алгоритмов генерации высококачественного кода.

Второй создавался в Новосибирском филиале ИТМиВТ (под руководством И. С. Голосова) одновременно и для Эльбруса 3 и для МКП. Входной язык транслятора включал помимо традиционных диалектов Фортрана некоторые конструкции Fortran 90. Особое внимание уделялось оптимизации программ, векторизации, компактификации кода. Большая часть транслятора одинакова для обеих архитектур и лишь при окончательной генерации варианты его различны.

В настоящее время продолжение работ по проекту Эльбрус под вопросом, равно как и завершение указанных систем. Обе группы сейчас активно работают с Sun Microsystems, что, может быть, позволит использовать (хотя бы частично) полученные наработки.

**2.11.** С созданием систем программирования для проекта Эльбрус связана и существующая в лаборатории технологии программирования и экспертных систем Института математики и механики Санкт-Петербургского университета группа под руководством В. О. Сафонова. Группа была организована более пятнадцати лет назад С. С. Лавровым для поддержки широких планов создания программного обеспечения для Эльбруса.

В этой группе разработан широкий спектр систем программирования для Эльбруса — реализованы языки Паскаль, Модула-2, CLU, SNOBOL-4, Рефал, ABC (язык символьной обработки, предложенный С. С. Лавровым), а также ряд интерактивных систем программирования — для языков Алгол 60, Basic, Forth.

Уже из перечисления языков программирования, реализованных этой и двумя предыдущими группами, видно, насколько амбициозным (в положительном смысле слова) был проект Эльбрус. В конечном счете для Эльбруса (не только в этих группах) были созданы трансляторы для всех мало-мальски употребляемых и для ряда экспериментальных языков, причем целенаправленно, в рамках общегосударственной программы.

В реализованных в этой группе трансляторах использован ряд новых и интересных решений по трансляции, например в трансляторе с CLU. На основании этих разработок и в их поддержку была создана так называемая TIP-технология (Technological Instrumental Package), предназначенная для создания больших модульных систем [12]. TIP-технология предполагает более содержательную спецификацию пакета как элемента сложной системы, чем это существует в известных языках. Достаточно детально регламентируется интерфейс пакета, что облегчает его переиспользование. На основе TIP-технологии был разработан ряд трансляторов, причем с использованием одних и тех же пакетов в различных трансляторах.

В настоящее время группа, насчитывающая около 10 сотрудников, разрабатывает транслятор с Паскаля для Эльбруса-3. Так же, как и предыдущие две, группа сотрудничает с Sun Microsystems и ведет разработку Паскаль-транслятора для новой рабочей станции семейства SPARC.

**2.12.** Одним из известных российских коллективов, ведущих разработки и исследования в области трансляторов и окружений программирования, является лаборатория системного программирования Института математики и механики Санкт-Петербургского университета (руководитель лаборатории — А. Н. Терехов).

Коллектив был организован Г. С. Цейтиным и был связан с созданием одной из наиболее известных в России и мире системы программирования с языка Алгол-68. Мало того, что реализация оказалась успешной, но эта группа стала активным центром по внедрению Алгола-68 в реальную практику программирования. Этот интересный (тем более ко времени своего создания) язык, как показала международная конференция, посвященная 25-летию его создания (Амстердам, февраль 1993 г.), стал к настоящему времени уважаемым, но практически мертвым (подобно латыни) языком — везде, кроме области влияния данной группы. Ее деятельность показала практическую мощь и хорошее соответствие языка большому кругу прикладных задач.

В настоящее время коллектив насчитывает около 60 сотрудников. Основное направление деятельности — создание технологии разработки программ для систем реального времени, прежде всего, для систем связи [13]. Основу создаваемой технологии составляют языки высокого уровня, обеспечивающие большую надежность разрабатываемого программного обеспечения, а именно языки с полным статическим контролем типов (Алгол-68, Модуль-2, Ада). Они, и в первую очередь Алгол-68, положены в основу большинства создаваемых технологических инструментов. В коллективе созданы трансляторы для ряда подобных языков, наиболее интересными из которых являются трансляторы для Алгола-68 и Ады.

Наиболее важной для технологии, создаваемой в коллективе, является система программирования WBC [14] для Алгола-68. Входной язык системы учитывает ряд новых предложений по расширению языка, связанных с модульностью, отдельной трансляцией, обработкой исключительных ситуаций.

Отличительной особенностью системы является ее ориентация на несколько инструментальных и целевых машин. Вся структура трансляторов и динамических окружений организована таким образом, чтобы выделить машинно- и операционно-зависимые части и унифицировать обращение к ним. Все трансляторы системы WBC одинаковы по синтаксису промежуточных языков: спецификации структур данных и доступа к ним; способам распределения памяти и регистров; технике генерации объектного кода; спецификации процедур динамической поддержки; способам выбора вариантов реализации конструкций языка. В трансляторах существуют единые программы монитора интерфейсной системы, видонезависимого и видозависимого (в терминологии Алгола-68) анализа, фрагментов фазы оптимизации, текстового редактора и отладки.

Специфика проблемной области — высокие требования к эффективности и надежности создаваемого по данной технологии программного обеспечения. Поэтому трансляторы системы обладают весьма изощренной техникой квазилокальной оптимизации программ, связанной прежде всего с развитым casing при реализации конструкций языка — особое внимание при этом уделяется тем конструкциям, реализация которых в наибольшей мере влияет на эффективность объектного кода (циклы, вырезки, процедуры). Система содержит средства конфигурационного контроля и поддержки разработки многомодульных программных комплексов.

Система интегрирована с другими средствами инструментальной поддержки разработки, в качестве которых выступают средства планирования и управления ходом разработки проекта, инструменты текстового и графического проектирования SDL-диаграмм (известный язык спецификации систем связи), средства комплексирования, отладки и измерения программ, инструменты вероятностного и имитационного моделирования аппаратуры и среды, ассемблеры и интерпретаторы целевых ЭВМ, действующие на инструментальных ЭВМ и пр. Разработаны специальные языки для описания оборудования и связей между его компонентами, а также процессоры, которые преобразуют эти описания в специальную базу данных, используемую при создании программного обеспечения средств связи.

Работа над технологией разработки программ, основанной на статических языках, привела к созданию архитектуры ЭВМ Самсон, ориентированной на программы, получаемые трансляцией с этих языков. В настоящее время Самсон выпускается как в виде отдельного устройства (процессор и локальная память), так и в виде троированного комплекса высокой надежности.

Самсон — микропрограммируемая ЭВМ. Средства микропрограммирования для нее разработаны на основе языка Алгол 68 и позволяют на порядок облегчить микропрограммирование по сравнению с традиционными микроассемблерами, что делает возможной микропрограммную эффективизацию конкретных программ. Сейчас на основе этого ведется проработка методики настройки ЭВМ Самсон на заданные предметные области.

Для языка Ада осуществлена полная реализация его и средств поддержки программирования в виде интегрированной инструментальной среды Pallada. Последняя содержит набор инструментов для разработки Ада-программ, в том числе отладчик в терминах входного языка, собственный текстовый редактор, систему конфигурационного контроля. Особое внимание в системе уделяется поддержке многобиблиотечного окружения. Специально для технологических целей в систему введены такие возможности, как гибкое управление проектом на основе абстрактных атрибутов, связанных с разделами библиотек: как поддержка запросов программ из библиотек, аналогичных принятым в базах данных. Система обладает высокой скоростью компиляции и перекомпиляции, в частности, за счет автоматической минимизации необходимых перекомпиляций, вызванных изменениями в контексте, и реализации пошагового связывания (incremental binding).

Pallada реализована для IBM/370 под VM/SP и PC 386 под UNIX. Система в принципе доведена до состояния программного продукта, однако не имеет использования по тем же причинам, что и описанная в п. 2.4.

**2.13.** Ряд исследовательских групп Санкт-Петербурга ведут работы, в области трансляции и построения окружений программирования, основанные на языке Форт.

В Санкт-Петербургском институте информатики и автоматизации РАН работы, ведущиеся под руководством С. К. Баранова, связаны с использованием Форты как инструментального средства для создания системного и прикладного программного обеспечения и на переносимой реализации этого языка, — Бета-Форт, — разработанной С. Н. Барановым. Создан программный пакет, выполняющий функции системы построения трансляторов и ориентированный на реализацию специализированных языков для приложений. Пакет состоит из форт-определений, к которым для конкретного языка добавляются форт-определения, отражающие синтаксические правила (в виде регулярных выражений) с включенными обращениями к семантическим (генерационным) процедурам, описания которых задаются также форт-определениями. Трансляция такого конкретизированного пакета порождает транслятор для данного специализированного языка (его шитый код). Существенно, что указанный пакет легко комбинируется с другими компонентами окружения программирования, также реализованными на языке Форт, и может настраиваться на конкретный специализированный язык, Такими как

пользовательский интерфейс, управление файлами, системы меню и протоколирования работы, средства отладочного слежения и пр. Подобный подход к построению трансляторов и окружений был опробован на ряде специальных языков и позволял создавать эффективные их реализации.

В Санкт-Петербургском техническом университете под руководством В. П. Котлярова разработана технология построения программного обеспечения для встроенных ЭВМ (технология КОМФОРТ), основанная на модульном расширении языка Форт. Были созданы инструменты этой технологии, поддерживающие сборочное программирование. Группой велись исследования по применению методологии смешанных вычислений для построения эффективного программного кода.

### 3. Промышленная разработка трансляторов и систем программирования

Большинство приведенных в п.2 примеров разработок трансляторов и систем программирования больше характеризовало исследовательский уровень разработок, чем их промышленное качество. Отсутствие реального рынка и конкуренции программных продуктов привело к тому, что даже широко использовавшиеся трансляторы не соответствовали международным требованиям на программный продукт, а притязания разработчиков прежде всего заключались в том, чтобы сделать "интересный" транслятор, а не транслятор, хорошо соответствующий таким пользовательским потребностям, как удобство работы, хорошее окружение, достаточная простота освоения и пр.

Следует заметить, что рынок отечественного программного обеспечения, в частности трансляторов, до сих пор в России не сложился, а следовательно, и реального промышленного производства не существует. Тому есть несколько причин:

- большая часть программного обеспечения, в первую очередь системного, к которому относятся и трансляторы, до сих пор распространяется нелегально, за счет несанкционированного (и практически бесплатного для собственников ПО) копирования,
- не существует системы продаж и сопровождения для отечественного программного обеспечения (и даже нет традиций организации такой системы). Складывающаяся сейчас система обязана своим возникновением зарубежным производителям ПО, которые не имеют никакого отношения к отечественным производителям и не заинтересованы в их работах,
- доведение системы программирования до конкурентноспособного программного продукта требует больших финансовых вложений, окупаемость которых на внутреннем рынке сейчас непредсказуема.

Несмотря на общую пессимистичность оценки следует отметить, что существуют и положительные доводы в пользу отечественных разработчиков трансляторов. Есть ряд систем, которые могли бы практически продаваться --- это Mithrill (см. п. 2.5), система WBC (п. 2.12), Pallada (п. 2.12). Хороший программный продукт могли бы представлять собой после определенного доведения Setl и Рефал-системы (п. 2.6 и 2.1 соответственно).

Важно отметить, что квалификация отечественных групп, работающих в области трансляции, весьма велика и они способны создавать трансляторы с высокими функциональными свойствами (скорость трансляции, качество объектного кода). Анализ ряда широко распространенных зарубежных систем программирования показывает, что при своих высоких внешних потребительских свойствах по функциональным свойствам они не выделяются на фоне отечественных систем.

Чтобы быть правильно понятым, хочу сказать, что многие из перечисленных групп имели достаточный опыт в широком распространении созданных ими систем, но отечественные требования все-таки не соответствовали современным международным стандартам к распространяемому рыночному продукту — ни по качеству документации, ни по технике выпуска версий программного продукта (что требует высокой технологичности как разработки самого продукта, так и подготовки следующих версий).

Следует отметить, что ряд из перечисленных в п.2 групп умеют (и намереваются) создавать полупромышленный продукт — такие системы программирования, которые достаточно широко используются или будут использоваться для создания больших программных проектов в

организациях, связанных с разработчиками. Это и система RAMPA (п. 2.1), и СОКПАТ (п.2.5), и инструменты, создаваемые в Санкт-Петербурге (п. 2.12). Заслуживает внимания путь, создания трансляторов для FSF, который реализуется в группе В. Иванникова (п. 2.3). Возможна организация центров отечественной промышленности трансляторов при содействии зарубежных фирм, что соответствует деятельности групп, перечисленных в п. 2.10 и 2.11.

#### 4. Заключение

Российские работы в области трансляции могут быть интересны извне в трех отношениях:

1. Включение российских организаций в исследования, ведущиеся в зарубежных научных центрах. Большинство перечисленных здесь результатов и оригинальны и интересны. Взаимосвязь отечественных и зарубежных научных организаций несомненно будет полезна обеим сторонам.
2. Использование уже существующих в России наработок для производства промышленных трансляторов для международного рынка.
3. Прямое приспособление работ по технологиям разработок ПО для встроенных ЭВМ и систем реального времени (силами российских разработчиков для отечественных и зарубежных фирм, связанных с подобными программными проектами).

#### Литература

- [1] Nepomiastchy P. "Rapport sur l'etat de la recherche, du development et de l'industrialization du logiciel en Russie" // Rapport de l'INRIA, fevrier 1994, 72 p.
- [2] Krukov V.A., Pozdnjakov L.A., Zadykhailo I.B. "RAMPA — CASE for portable programs development" // Parallel Computing Technologies (РАСТ-93), —Obninsk, 1993.
- [3] Серебряков В.А. "Основные особенности входного языка и реализации СПТ СУПЕР" // Программирование, 1982, No.1, с.78-88.
- [4] Серебряков В.А., Бездушный А.Н, Белов К.Г. "Синаис-3 — расширение языка Си для параллельных вычислений при решении научных задач" // Программирование, 1994, No.2, с.3-15.
- [5] Непомнящий В.А., Шилов Н.В. "REAL 92: комбинированный язык спецификаций для систем и свойств взаимодействующих процессов реального времени" // Программирование, 1993, No.6, с.64-80.
- [6] Касьянов В.Н., Сабельфельд В.К. "Инструменты преобразования программ" // Автоматизированное рабочее место программиста. — Новосибирск, 1988, с.4-12.
- [7] Поттосин И.В. "СОКПАТ — система окружения программирования для встроенных ЭВМ" // Новосибирск, 1992, 18 с. — (Препр. РАН Сиб. отделение, ИСИ; No.11).
- [8] Nedorya A, Nikitin A. "Mithril — portable Oberon-2 environment" // Modular Magazine, 1993, No.1, с.8-14.
- [9] Mirenkov N.N. "The Siberian approach for an open-system high performance computing architecture" // Computing & Control Eng., I, 1992, No.3.
- [10] Mantsivoda A. "Flang and its Implementation" // Lect. Notes Comput. Sci. 1993, V.714, p.151-161.
- [11] "Язык представления знаний РЕПРО" // Артемьева И.Л., Клещев А.С., Лифшиц Ф.Я., Плис Г.Я. // Объектно-ориентированное программирование: Тез. докл. Всесоюз. конф. "Актуальные проблемы системного программирования". — Таллин, 1990, с.84-86.

- [12] Сафонов В. О. "Языки и методы программирования в системе "Эльбрус". — М.: Наука, 1989, 392 с.
- [13] Терехов А. Н. "Технологические средства программирования — методы и инструменты" // Информатика и программирование, Новосибирск, 1989, с.5-16.
- [14] Лаврова Ю. К. "АЛГОЛ-68 для IBM PC" // Информатика и программирование, Новосибирск, 1989, с.121-124.
- [15] Поттосин И. В. "Текущее состояние российских исследований и разработок в области трансляции" // Новосибирск, 1995, 32 с. — (Препр./РАН Сиб. отделение, ИСИ; No.30).

---

**Об авторе.** Игорь Васильевич родился 21 февраля 1933 г. в селе Кинель-Черкассы Самарской области. В 1955 г. окончил "специальный факультет" Томского университета по специальности "баллистика" и получил распределение в одну московскую военную организацию. В 1958 г. перешёл в Институт математики с вычислительным центром только что возникшего Сибирского отделения АН СССР, базировавшийся ещё в Москве, где стал одним из ближайших сотрудников молодого тогда Андрея Петровича Ершова. Прославившаяся в дальнейшем тройца — А. П. Ершов, Г. И. Кожухин, И. В. Поттосин — явилась ядром Отдела программирования ВЦ СО АН СССР. Когда не стало Андрея Петровича, Игорь Васильевич стал его преемником и символом продолжения духа и традиций Отдела, Сибирской школы программирования. В 1990 г. Отдел перерос в Институт систем информатики СО РАН, директором которого с 1992 по 1998 гг. был Игорь Васильевич. Основной областью научных исследований И. В. Поттосина в последние годы были инструментальные системы поддержки информационных технологий и разработка прототипов такой поддержки. 15 декабря 2001 г. Игоря Васильевича не стало, он не дожид до своего 70-летия чуть больше года. "Это бесконечно печально. Уходят пионеры, и какие люди. Невосполнимо: где замена на должном уровне культуры, глубины, знаний?" (Л. А. Калиниченко, Институт проблем информатики РАН).