

47. *Винер Н.* Кибернетика, или управление и связь в животном и машине. — 2-е изд. — М.: Сов.радио, 1968. — 326 с.
48. *Кухтенко А.И.* Кибернетика и фундаментальные науки. — Киев: Наук. думка, 1987. — 142 с.
49. *Кухтенко А.И.* На пути к единству научных знаний // Методологический анализ физического познания. — Киев: Наук. думка, 1985. — С.130—146.
50. *Кухтенко О.И., Самойленко Ю.И.* Автоматичне керування плазмовими об'єктами. — Вісн. АН УССР. — 1971. — № 3.
51. *Комбинированная система управления плазменным шнуром / А.И.Кухтенко, Ю.И.Самойленко, Ю.П.Ладиков-Роев, В.К.Бутенко // Там же.*
52. *Kukhtenko A.I., Samoilenko Yu.I.* Stabilisiertes Plasma für die Kontrollierte Kernfusion // Ideen des exakten Wissens. — 1973. — № 7. — S. 413—422.
53. *Самойленко Ю.И.* Проблемы и методы физической кибернетики // Праці Інституту математики НАНУ. — 2006. — Т. 56. — 644 с.
54. *Бутковский А.Г., Самойленко Ю.И.* Управление квантово-механическими процессами. — М.: Наука, 1968. — 256 с.
55. *Гуленко В.П.* Работы А.И. Кухтенко по авиационным тренажерам // Проблеми інформатизації та управління. — 2004. — № 11. — С.35—37.

Получено 25.10.2007

А.М. Глебова, Т.О. Кухтенко

О.И. Кухтенко та його наукова школа

Розповідається про життєвий шлях, наукову та педагогічну діяльність академіка НАН України О.І. Кухтенка (1914—1994). Проаналізовано його праці та праці членів створеної ним наукової школи. Наведено спогади учнів, колег та дочки О.І.Кухтенка, що відтворюють його постать як ученого, вчителя, людини.

О.Л. Перевозчикова

Школа теории программирования Е.Л. Ющенко

Обзор посвящен истории возникновения и сегодняшнему дню школы теоретического программирования, до начала нового тысячелетия (в течение 40 лет) работавшей под руководством члена-корреспондента НАН Украины Е.Л.Ющенко на базе созданного ею в 1957 году отдела автоматизации программирования Института кибернетики имени В.М.Глушкова НАН Украины. С позиций сегодняшнего дня сделана попытка оценить работы Е.Л.Ющенко и ее ближайших соратников и особенно учеников — бывших аспирантов: докторов физ.-мат. наук, профессоров И.В.Вельбицкого, Е.М.Лаврищевой и А.И. Халилова, доктора техн. наук, профессора Г.Е.Цейтлина, чл.-кор. НАН Украины, доктора физ.-мат. наук, профессора О.Л.Перевозчиковой.

На Украине теория программирования (ТП) начала развиваться с 1957 года, со дня основания ВЦ АН УССР, реорганизованного впоследствии в Институт кибернетики АН УССР. Основной предмет исследований ТП — это языки программирования, их инструментальные среды и системный интерфейс для разработки программного продукта. Основа математического ап-

парата — компьютерная алгебра и логика, теория автоматов и формальных языков, алгебро-грамматический аппарат спецификации классов алгоритмов и программ, формальные модели представления знаний. На основе математического аппарата разработаны модели и методы представления знаний и инструментарий конструирования компиляторов, интерпретаторов, средств

© О.Л. Перевозчикова, 2007

общесистемного назначения, баз данных и знаний, экспертных систем, систем принятия решений и профессиональных систем для различных предметных областей (приложений).

Существенное развитие ТП справедливо связывают с работами чл.-кор. НАН Украины Е.Л.Ющенко, создавшей в Институте кибернетики (ИК) научную школу в этой области. Под научным руководством Е.Л.Ющенко выполнены важные работы по ТП, разработаны разнообразные системы обработки данных, трансляторы языков программирования на компьютерах разных типов, заложены научные основы современной ТП. С позиций сегодняшнего дня остановимся на основных работах Е.Л.Ющенко и ее учеников и применении этапных достижений.

Выдающийся ученый-математик, специалист по программному обеспечению компьютеров, член-корреспондент НАН Украины, заслуженный деятель науки Украины, Екатерина Логвиновна Ющенко (8.12.1919 — 15.08.2001) стоит вместе с В.М. Глушковым в первом ряду основателей Института кибернетики, с которым она навсегда связала свою жизнь.

Научную деятельность в системе Академии наук УССР Е.Л. Ющенко (Рвачева) начала в августе 1946 года во Львовском отделении Института математики. Первый цикл ее исследований относится к теории вероятностей и математической статистике. Здесь она получила ряд оригинальных результатов по многомерным устойчивым законам распределения, локальным предельным теоремам и критериям сравнения эмпирических данных. Как оппонент ее кандидатской диссертации академик А.Н. Колмогоров высоко оценил эти результаты, нашедшие применение в квантовой механике и в настоящее время ставшие классическими.

Становление научной школы. В 1951 году с введением в действие первого отечественного компьютера МЭСМ воз-



никала насущная потребность дальнейшего развития компьютерной техники. Работавшая тогда ученым секретарем Института математики АН УССР Е.Л.Ющенко со своими учениками и соратниками (Ю.В. Благовещенский, И.В. Погребыцкий, В.С. Королук, Ю.А. Митропольский, М.Р. Шура-Бура, Ю.Д. Шмыглевский, А.А. Ющенко и др.) переключилась на разработку алгоритмов решения классов народнохозяйственных задач методами классической и дискретной математики, которые быстро развивались.

С созданием в Киеве МЭСМ государство определило главное ее назначение — расчеты ядерного оружия и вычисления баллистики ракет и параметров их боеголовок. Как и большинство советских естественных наук, кибернетика должна была работать прежде всего на военно-промышленный комплекс. Именно об этих первых решенных задачах, поставленных президентом АН СССР М.В.Келдышем, часто вспоминала Е.Л.Ющенко. Достигая этой цели, чиновничья элита СССР стремилась добиться скорейших результатов, поэтому прежде всего финансировались

стратегически значимые задачи разработки новой компьютерной техники и методов прикладной математики. Причем НИР по программной составляющей ИТ рассматривались лишь как производные от этих задач, скупо финансировались, а результаты недооценивались.

Сегодня трудно вообразить атмосферу 50-летней давности, когда кибернетика, считавшаяся буржуазной лженаукой, по сути возникла на голом месте. Зарождение ТП и оформление ее начальных результатов в первых языках программирования стали следствием величайшего изобретения середины XX века — компьютера. Нельзя обойти вниманием огромный интерес к программированию в те годы. Профессия программиста была окружена ореолом престижности и стала символом того времени подобно тому, как профессия шофера ассоциируется сейчас с началом XX века.

Ограниченность внутренней памяти МЭСМ (100 ячеек оперативной и 64 ячейки односторонней сменно-наборной с разрядностью 21 бит), слабое быстродействие (100 операций/с), неустойчивость работы, вызванная реализацией МЭСМ на электронных лампах числом до 5000 штук, вынуждали составителей программ изыскивать изощренные способы использования внутреннего языка компьютера. Составление каждой программы рассматривалось как решение индивидуальной задачи. Программисты находили экономные решения, искусно используя систему команд компьютера; устраивали своеобразные соревнования по улучшению отдельных программ.

Короткая программа была обусловлена не столько ограниченностью оперативной памяти, сколько тогдашним способом общения человека с компьютером. Перед запуском программы ее каждый раз требовалось вручную ввести в оперативную память с помощью двух десятков тумблеров на специальной панели. Чтобы через несколько се-

кунд или минут после запуска программы увидеть на табло результат решения задачи, требовалось по крайней мере полчаса поработать пальцами, переводя тумблеры в соответствующее положение и задавая таким образом (в бинарном коде!) каждую команду программы. Причем программисты вводили программу в оперативную память МЭСМ по двое или по трое, чтобы убедиться в правильности ввода и в случае аварийного завершения искать ошибку в самой программе или входных данных. Напомним, что первые дисплеи стали использовать для взаимодействия с компьютером только после 1972 года.

Возникла новая профессия — оператор ЭВМ, который помогал программистам воплощать их алгоритмы на компьютере. Программисты мечтали об удобных средствах символьного ввода-вывода, а им приходилось много усилий тратить на подготовку сначала перфолент, потом перфокарт, повышение надежности сохранения данных на постоянно слипавшихся магнитных лентах и выдумывание схем распечатки результатов на крайне несовершенных принтерах.

По прочтению этих строк у молодых ИТ-специалистов обязательно возникнет вопрос, почему пользователи МЭСМ с неподдельным энтузиазмом работали на таком несовершенном инструменте. В 1957—1960-х годах в отделе Е.Л.Ющенко, располагавшемся рядом с МЭСМ в трапезной Феофаниевского монастыря под Киевом, работало до полусотни так называемых математиков-вычислителей, которые по 45 часов в неделю, используя арифмометры “Рейнметалл”, вручную рассчитывали задачи по весьма сложным процедурам, оформленным в ПОЛИЗе (Польской инверсной записи Лукасевича). Эта изнурительная работа требовала полной самоотдачи, иначе неизбежные ошибки в расчетах могли уничтожить плоды многодневного труда. Для повышения достоверности расчетов одну задачу решали параллельно не-

сколько вычислителей, регулярно сверяя промежуточные результаты и в случае их несовпадения возвращаясь на предыдущие контрольные точки. На этом фоне затраты от ввода в МЭСМ исполняемого кода программы были мизерными, а программу можно было запускать многократно и за весьма короткий срок получать искомое решение для достаточно широкого диапазона входных данных. Когда же для ввода программ и их входных данных задействовали первые носители информации, все окончательно отдало предпочтение МЭСМ для организации расчетов.

Кроме задач внешней баллистики, к числу задач, программы решения которых составили математическое обеспечение первого отечественного компьютера, вошли:

- ❖ составление таблиц для статистического приемочного контроля (постановка задачи Б.В. Гнеденко, исполнитель Е.Л. Ющенко);
- ❖ динамические задачи теории упругости (постановка А.Ю. Ишлинского, Институт математики АН УССР, исполнитель А.А. Ющенко);
- ❖ выбор оптимальных параметров шахтных канатов (постановка Г.И.Савина и А.Ю.Ишлинского, исполнитель А.А. Ющенко);
- ❖ определение областей устойчивости энергосистем, в частности Куйбышевской ГЭС (постановка Л.В. Цукерника, Институт электротехники АН УССР, исполнители В.С.Королюк, Е.Л.Ющенко);
- ❖ расчет тепловых напряжений строительных конструкций (постановка А.Д.Коваленко, Институт механики АН УССР, исполнитель Е.Л.Ющенко);
- ❖ обработка геодезических наблюдений (постановка Н.И.Якубецкой, исполнитель Е.Л.Ющенко);
- ❖ расчет задач синтеза аммиака (исполнитель Л.Н.Иваненко);
- ❖ оценка объемов земляных работ при проектировании автодорог (постановка А.К. Хавкина, Киевский авто-

дорожный институт, исполнители Е.Л. Ющенко, Л.Н. Иваненко, А.М. Сибирко).

Для более сложных задач выяснилось, насколько затруднительно решать их путем кодирования машинных программ согласно определенной программистом блок-схеме алгоритма, поэтому для общения человека с компьютером возникла проблема создания языка программирования высокого уровня и соответствующего компилятора. Тогда речь шла о новом применении компьютера для решения задач неарифметического характера (аналитические преобразования выражений, их дифференцирование и интегрирование, распознавание графических образов, обработка текстов и со временем проверка синтаксической правильности программ). Соответственно сформировался предмет исследования ТП — методы и средства разработки программ, в настоящее время их называют программологией. А в 1960-е годы эти средства отождествляли с процедурными языками, воспринимаемыми как знаковые системы общения с компьютером. Заметим, что сначала языки называли алгоритмическими по аналогии с алгоритмическими системами (например АЛГОЛ-60 задуман как язык записи алгоритмов, не имевший средств ввода-вывода) и, лишь осознав всю неформализованность операционной среды для разработки и функционирования программ, перешли к названию “языки программирования”.

Одну такую алгоритмическую систему в качестве операторного метода программирования предложил в 1952—1953 гг. чл.-кор. АН СССР А.А. Ляпунов. Здесь единицы действия — операторы текста программы — классифицированы по функциональности: операторы присваивания значений арифметических выражений воздействовали на данные, логические операторы группировали из вычислений логических отношений и передач управления, а операторы модифика-

ции влияли на другие операторы через специальный параметр, допуская инициализацию, переадресацию на величину (возможно, приращенного) параметра и восстановление начального вида оператора. Каждая программа состояла из двух частей. Схема программы как символическое представление операторов отмечала передачи управления и классифицировала операторы, а во второй части в спецификации операторов описывали их конкретное содержание (семантику). Поэтому в тексте программы нашли отражение одновременно два этапа разработки алгоритма: его общий план с конкретизацией, творчески разрабатываемые программистом, и затем рутинное кодирование на машинном языке, которое можно было довольно просто формализовать и отделить от разработчика алгоритма. Сразу определилась задача семантических эквивалентных преобразований схемы программы с целью ее оптимизации; это привело к теории схем программ, которая в течение 1965—1980 гг. определила направление исследований новосибирской школы ТП во главе с академиком А.П.Ершовым. Кроме того, стала понятна задача разработки компилятора, который тогда называли программирующей программой.

Существенное влияние на понимание проблем ТП для развития исследований киевской школы ТП оказал профессор Киевского университета Л.И. Калужнин. Вернувшись из парижской Сорбонны на Родину, он читал в 1950—1970-е годы курс математической логики и предложил формальный аппарат граф-схем программ.

Определяющий этап исследований в научной деятельности Е.Л. Ющенко и ее школы ТП связан с понятием *адресного программирования*, на языковой основе воплотившего два общих принципа работы компьютера — адресности и программного управления. Создавая удобную систему понятий для описания архитектуры компьютера и его системы команд, Е.Л.Ющенко совместно с акаде-

миком НАН Украины В.С.Королюком ввели в адресный язык средства манипулирования *адресами второго ранга*, вошедшими в современные языки как классические конструкции pointer.

Именно создание в 1955 г. адресного языка — первое фундаментальное достижение всей киевской школы ТП. Опередив создание первых языков программирования ФОРТРАН (1958), КОБОЛ (1959) и АЛГОЛ-60 (1960), адресный язык предвосхитил появление не только языков программирования с аппаратом косвенной адресации, но и ассемблеров. В 1961—1965 гг. учебники по адресному языку изданы в пяти странах на русском, словацком, венгерском, немецком и французском языках. Разработка и внедрение серии компиляторов с адресного языка на всех отечественных компьютерах первого поколения “Киев”, “Днепр”, “Урал”, “Минск”, М-20 во многом содействовали подъему уровня программирования в СССР и за его пределами. Причем первый отечественный компилятор заработал в 1962 году в Киеве для машины “Урал-1” и имел входной адресный язык, а в Москве и Новосибирске компиляторы с АЛГОЛА-60 появились тремя годами позднее.

Механизм косвенной адресации аппаратно реализовали на компьютере “Киев” и это едва ли не первый пример влияния ТП на проектирование архитектуры и элементной базы компьютеров. Здесь уместно сослаться на воспоминания Е.Л.Ющенко о появлении такого механизма и осознании его роли в программировании. Поскольку упомянутых 100 ячеек оперативной памяти МЭСМ катастрофически не хватало, программисты придумали оригинальный прием расширения ресурса за счет так называемой “сменно-спаянной” памяти, т.е. набора плат (напомним, на лампах), на которых были аппаратно зашиты распространенные функции и операции: логарифмирования, извлечения корней, тригонометрические функции и т.п. Если в программе требова-

лось обратиться к элементу такого набора, то программировали останов и на панели высвечивался номер, согласно которому нужно было вручную вставить соответствующую плату. После вычисления нужной функции место платы (слота) можно было использовать для другого элемента набора. Так зародилась концепция стандартных библиотек функций, которая оформилась в языках программирования (в нашем случае в адресном языке) позднее, в начале 60-х годов благодаря работам В.М. Глушкова и его аспиранта А.А. Стогния. Они разработали метод специализированных программ, во многом аналогичный методу крупноблочного программирования лауреата Нобелевской премии, академика АН СССР Л.В. Канторовича. Этот метод с 1954 года развивался в поддержку процедур компьютерной балансировки государственных планов (пятилетних, ежегодных, отраслевых, межотраслевых) развития СССР.

За короткий срок разработки киевской школы ТП получили признание среди ведущих специалистов по программированию. Основанная в начале 1950-х годов и возглавляемая Е.Л. Ющенко киевская школа ТП уже с начала 1970-х годов с расширением тематики дала жизнь самостоятельным научным школам, которые возглавили ученики и соратники Е.Л. Ющенко — академики НАН Украины И.В. Сергиенко и В.Н. Редько, чл.-кор. РАН и НАН Украины А.А. Стогний, проф. И.В. Вельбицкий.

Языковые процессоры 1960-х годов. Усилиями Е.Л. Ющенко и д-ра физ.-мат. наук Т.А. Гринченко один из первых советских компиляторов заработал в 1962 г. в Киеве для компьютера “Урал-1” (сразу затем для ЭВМ “Киев”) и имел входной адресный язык. В этом языковом процессоре впервые в отечественной практике задействована продуктивная однопроходная схема компиляции, а для обработки выражения использован ПОЛИЗ. За короткое время реализовали семейство компиляторов адресного язы-

ка на разных компьютерах, а теоретические достижения этой реализации стали предметом докторской диссертации Е.Л. Ющенко, оказавшейся в 1965 г. первым в СССР доктором физико-математических наук по программированию.

К тому времени для операторного метода А.А. Ляпунова заработало несколько компиляторов на московских компьютерах БЭСМ-2 и “Стрела”, преимущественно с многопроходными схемами обработки текста программ, требовавшими дополнения специфическими операциями для обеспечения (например как в ПП-2 под руководством проф. М.Р. Шуры-Буры) рационального размещения данных в оперативной памяти и компактности программы. Уже выяснилась функциональность таких служебных подсистем, как загрузчики программ, линкователи (в современной терминологии утилиты раннего связывания программ согласно модульному принципу их разработки), отладчики (дебагеры), сегодня представляющие важное звено инструментальных сред разработки программ и совершенно незаметно для пользователя распределяющие информационные ресурсы, контролирующие поведение программ в ходе вычислений, имитирующие прокручивание с выдачей оперативной информации и “посмертных” сведений и т.п.

Напомним, что в 1960-е годы сложилась парадоксальная ситуация, когда автоматизация программирования благодаря реализации компиляторов опередила разработку операционных систем с их необходимой составляющей символического кодирования на языках ассемблеров и применением библиотек для накопления компонентов повторного использования. Ситуация немного выправилась в 1964—1966 гг. с разработкой компиляторов ТА-1, ТА-2, “Альфа” с языка АЛГОЛ-60 для компьютера М-20, воплощенного в нескольких крупных промышленных сериях М-20, БЭСМ-3М, БЭСМ-4, М-220, М-222. Во-первых, накопленный отечествен-

ный опыт обогатился результатами американских и немецких разработчиков, отношения с которыми стали доступнее. Во-вторых, сам язык АЛГОЛ-60 завершил определенный этап формирования алгоритмики процедурных языков и зафиксировал отработанный за 1950-е годы набор языковых конструкторов для представления широкого класса алгоритмов, поэтому программирование стало признанной международной научной сферой. Общий логико-алгоритмический уровень языка АЛГОЛ-60, особенно описание его синтаксиса в виде так называемых Бэкусовых нормальных форм, резко повысили уровень теоретических исследований для конструирования языковых процессоров.

Бесспорное признание амбициозной московской школой ТП удачной реализации семейства компиляторов адресного языка имело большое значение. Прежде всего были признаны удобство и прозрачность стековых схем компиляции таких сложных языковых конструкторов, как арифметическое и условное выражения, циклы, массивы, процедуры и т.п.. Собственно наличие у киевских компьютеров механизма косвенной адресации позволило упростить большинство алгоритмов языкового процессирования и уменьшить или совсем ликвидировать непроизводительные пересылки массивов данных, вследствие чего скомпилированный код становился короче и его можно было построить за один проход программы.

В первые советские компиляторы для операторного метода А.А. Ляпунова заложена идея компоновки скомпилированного кода из частиц, отвечающих отдельным операторам алгоритма. Эти компиляторы были многопроходными, т.е. многократно просматривали обрабатываемую программу. Существенно дальше продвинулись разработчики семейства компиляторов адресного языка, использовав модульный принцип программирования и схему управления компиляцией на основе разделения

входной цепочки на элементарные символы. К тому же много усилий требовалось для изучения схем промежуточной компиляции, в результате которой получали программу на другом языке. Таким образом повышалась мобильность программ, прежде всего самих программирующих программ, поскольку их жизненный цикл был довольно коротким ввиду частой модификации компьютерных платформ.

Эта ситуация напоминает современный этап развития распределенных приложений, функционирующих в гетерогенной сетевой среде. Именно на преодоление гетерогенности компьютерной сети задействован унифицированный байт-код, который виртуальная Java-машина интерпретирует, или декодирует поток байтов протокола GIOR для проведения маршалаинга типов данных в процессе соединения «клиент—сервер» согласно спецификациям OMG IDL. Речь идет о мобильности программ на уровне не их исходного кода (последовательность операторов некоторого языка программирования), а бинарного (исполняемого) кода.

Выдвинутые Е.Л. Ющенко идеи автоматизации процесса конструирования компиляторов путем их формального описания на уже реализованном языке программирования, а также концепция параметризации систем получили дальнейшее развитие в работах ее учеников и вылились в отдельные направления ТП и технологии программирования, получившие международное признание. Удачная попытка получения в 1962 г. на компьютере «Киев» компилятора адресного языка для другого компьютера «Днепр» почти на двадцать лет предвосхитила появление метакомпиляторов и генераторов компиляторов. А использование машины «Днепр-1» с развитой аппаратно зашитой библиотекой стандартных программ как управляющей ЭВМ в комплексе с набором датчиков на объекте и реализация ряда параметрических систем автоматизации производствен-

ных процессов (бессемеровский цех Днепропетровского металлургического завода, цех гальванопокрытий завода “Арсенал”, разметочно-маркировочный цех Николаевского судостроительного завода) продемонстрировали не только реальность алгоритмизации управленческих процессов, но и подлинность *автоматизации программирования*, на несколько порядков повысившей производительность труда программистов-разработчиков.

С 1963 г. одновременно с АЛГОЛом-60 распространились методы компиляции, основанные на синтаксическом управлении. Перевод с одного языка на другой в синтаксически управляемом языковом процессоре состоит из двух фаз: построение дерева грамматического разбора по записи исходной программы и замена вершин дерева их семантическими эквивалентами, описанными на выходном языке. При этом возникло много новых проблем, вызванных не самими языками, а грамматиками, задающими синтаксические правила этих языков. Поэтому конструирование синтаксически управляемых языковых процессоров, ориентированных на широкий класс языков и содержащих аппарат для своего расширения, стимулировало исследования по формализации синтаксиса и семантики языков программирования, основу которых составило представление семантики путем индукции по определенной синтаксической структуре языковых цепочек (программ). Отсюда возникла потребность в создании метаязыка описания синтаксиса и на его основе метаязыка для описания семантики языков программирования.

Заметим, что весомый вклад в достижения киевской школы ТП внесли разработчики семейства компьютеров МИР. Аппаратная реализация компиляторов для языков высокого уровня АНАЛИТИК (д-р техн. наук С.Б.Погребинский, доктора физ.-мат. наук В.П.Клименко и Т.А.Гринченко) помогла нейтрализовать недостаточность

вычислительных ресурсов компьютеров 1-го поколения. Именно в этом состоит одна из причин популярности и широкого внедрения семейства первых персональных компьютеров МИР и языков АНАЛИТИК со средствами аналитических (формальных) преобразований, положивших начало исследованиям по компьютерной алгебре.

Сложившийся уже в конце 1950-х годов стиль киевской школы ТП наиболее ярко характеризует выдвинутый В.М.Глушковым, Е.Л.Ющенко, Л.А.Калужниным автоматнo-алгебраический подход в программологии и затем в программной инженерии. Формально-логический характер исследований по алгоритмическим алгебрам, адресному языку и граф-схемам программ существенно повлиял на дальнейшие направления научных исследований и их результаты.

Возникновение языков спецификаций. Распространение АЛГОЛА-60 ознаменовало новую эру применения языков программирования для алгоритмизации разных предметных областей и существенного расширения круга пользователей компьютеров. Начали интенсивно разрабатываться языки спецификаций, тогда называемые проблемно-ориентированными, а соответствующие языковые процессоры — системами с персональной ориентацией. Цель таких языков и систем — максимально освободить пользователя от потребности в представлении алгоритма решаемой задачи и сохранить за ним лишь право на формулирование условий конкретной задачи, дабы универсальный алгоритм решения класса задач разрабатывали опытные специалисты. Классический пример — языки генераторов отчетов, использование которых сводится лишь к описанию формы нужного отчетного документа. Конечно, разработка систем с персональной ориентацией не могла не оказать содействие осмыслению форм общения с компьютером, тем не менее, ввиду отсутствия удобных устройств было еще далеко до

организации диалога через привычные сегодня экранные формы дисплея, с задействованием клавиатуры и разных устройств позиционирования курсора на экране.

Первые языки спецификаций и системы с персональной ориентацией реализованы в середине 1960-х, когда заработала управляющая ЭВМ “Днепр-1”. Использование этого компьютера с развитой аппаратно зашитой библиотекой стандартных программ в комплексе с набором датчиков на объекте и реализация ряда параметрических систем автоматизации производственных процессов (в том числе на металлургических комбинатах Берлина и г.Раквицы в тогдашней ГДР) продемонстрировали не столько реальность алгоритмизации управленческих процессов, сколько действенность автоматизации программирования, на порядки повысившей производительность работы программистов.

Позднее Е.Л.Ющенко приложила немало усилий для реализации и, главное, внедрения разных языков спецификаций. Сначала эти языки копировали форму конструкторов АЛГОЛа-60, потом появились другие образцы, среди которых нельзя обойти вниманием процедурный язык КОБОЛ, предназначенный для алгоритмизации банковской и кредитно-финансовой сфер. Тем не менее, в плановом хозяйстве СССР язык КОБОЛ не нашел применения, хотя компиляторы для него были довольно изысканно разработаны для нескольких семейств советских компьютеров. А проблемно-ориентированный язык ЯОД-75 стал полигоном распространения средств решения задач АСУ.

Пропаганда языка КОБОЛ имела оригинальные последствия. Когда в 1970-х годах началась первая волна эмиграции евреев из СССР, учебник по КОБОЛу стал бестселлером и вышел тремя тиражами, а сама Е.Л.Ющенко со своими сотрудниками стали консультировать своих знакомых, знакомых их знакомых и т.п. Поскольку знание

КОБОЛа гарантировало высокооплачиваемую работу банковского программиста, Е.Л.Ющенко неоднократно отмечала, что готовит персонал для зарубежных банков, но относилась к этому как научный сотрудник, отработывая на неожиданной прослойке пользователей алгоритмы языкового процессирования КОБОЛа.

Большое влияние киевская школа ТП и непосредственно Е.Л.Ющенко оказали на работы В.Л.Рвачева, который в 1970-е годы для решения задач прикладной математики и механики предложил класс R-функций как базисных функций для областей практически произвольной природы с различными вариантами краевых условий. Этот класс функций, который ныне ассоциируется с всемирно признанным методом RFM, возник на стыке математической логики, классических методов прикладной математики и современных методов кибернетики и составил основу входного языка семейства генераторов ПОЛЕ, реализованных соратниками В.Л.Рвачева и его учениками чл.-кор. НАН Украины Ю.Г.Стояном, докторами физ.-мат. наук Г.П.Манько, А.Н.Шевченко и другими для решения задач оптимального раскроя и особенно краевых задач математической физики.

Технология программирования 1980-х годов. В эти годы идея автоматизации процесса конструирования компиляторов путем их формального описания на уже реализованном языке программирования, а также концепция параметризации систем существенно повлияли на направления научных исследований и результаты разработки систем построения компиляторов и метакомпиляторов. В исследованиях киевской школы ТП выделились три основных направления:

1. Применение алгебраических методов в теории конструирования языковых процессоров и формализация на этой основе понятия языка программирования с последующим оформлением

в 1980-х годах концепции алгебраического программирования.

2. Формализация отдельных этапов построения языковых процессоров и создание параметрического подхода к их построению, основанного на организации синтаксического анализа по представлению так называемых беступиковых (беспереборных) грамматик, которые в сравнении, скажем, с Бэкусовыми нормальными формами имеют механизм регулярного управления выводом для однозначного выбора путей построения синтаксической структуры языковой цепочки (программы). Эту идею параметрического построения языковых процессоров конкретизировали в многоязыковых системах СТАРТ и Т-СЕМОЛ (версия адресного языка), разработавших на ЭВМ “Днепр-1”.

3. Применение теоретико-автоматных методов для представления синтаксиса и семантики языков программирования благодаря специальному метаязыку СМ-грамматик, который оказался наиболее удобным для разработчиков. Поэтому фронт приложений здесь был широчайшим, а опыт использования СМ-грамматик сформировал концепцию Р-технологии.

Высокой функциональности систем построения трансляторов и метакомпиляторов в значительной мере способствовало применение промежуточного языка-посредника, удобного для перевода с других языков. Впервые схему такого многоязыкового процессора испытывали на М-20 в двухходовом компиляторе с АЛГОЛа-60 и адресного языка, причем последний был еще и языком-посредником. Несмотря на механизм косвенной адресации, адресный язык уступал языку АЛМО, неоднократно успешно примененному московской командой М.Р.Шуры-Буры, благодаря его назначению как входного языка абстрактной вычислительной машины, воплотившей типичные черты наиболее распространенных к тому времени компьютеров.

В развитие результатов 1960-х и начала 1970-х годов усилиями Е.Л.Ющенко, А.А.Летичевского (направление 1), В.Н.Редько (направление 2), И.В.Вельбицкого (направление 3) и их учеников разработаны оригинальные классы грамматик для беспереборных синтаксических анализаторов (в современной терминологии — парсеров), что отвечало мировому уровню результатов. К тому времени в СССР лишь еще одна школа московского Института проблем управления, возглавляемая проф. Э.А.Трахтенгерцем, занималась такой проблемой, копируя грамматики предшествования западных разработчиков компиляторов.

К этим результатам киевской школы ТП пренебрежительно относились москвичи из ведущих учреждений АН СССР (ВЦ и Института математики им. Стеклова), не скрывая своего удивления, как киевляне могут изобрести что-то новое в том, что они доказали в общих теоремах математической логики и современной алгебры. Особое пренебрежение вызывали основополагающие работы В.М.Глушкова по системам алгоритмических алгебр; почти все москвичи-алгебраисты с докторскими степенями и едва ли не каждый кандидат наук считали эти результаты тривиальными и практически нецелесообразными. Как всегда, этот спор рассудило время, и уже в конце 1970-х годов за границей, а скоро и у нас заговорили о логическом программировании как следствии развития классических методов математической логики и алгебры. Только тогда акад. А.А.Самарский, возглавлявший школу прикладной математики в одноименном московском институте АН СССР, признал наличие определенного “сепаратизма программистов”, пользующихся собственным научным базисом информатики, который намного обширнее вычислительной математики.

В начале 1980-х годов разработка автоматически-алгебраических методов представления знаний и построение классов

распознавателей специалистами киевской школы ТП, опередив весь мир, привели к формулированию задач технологии программирования для модульно-сборочной генерации прикладных программ и систем, в решении которых кристаллизировались современные категории программной инженерии:

- ❖ инструментальная среда разработки программ;
- ❖ жизненный цикл программного продукта;
- ❖ концептуальное моделирование предметной области (домена);
- ❖ средства ведения программистского проекта;
- ❖ CASE-системы.

По инициативе Е.Л.Ющенко и при непосредственном ее участии разработаны СМ-грамматики и Р-технология И.В.Вельбицкого, АПРОП-сборочное программирование Е.М.Лаврищевой, “Мультипроцессист” Г.Е.Цейтлина, конфигурирование маршрутных систем О.Л.Перевозчиковой. Это первые практически результативные шаги в становлении программной инженерии, которую с конца 1980-х годов стали ассоциировать с коммерческими CASE-системами. Ряд инструментально-технологических средств не только придал инженерный характер процессу разработки программного продукта, но и нашел широкое применение в решении важных прикладных задач народного хозяйства, образования, науки и пр.

На разработанном для описания синтаксиса языков программирования аппарате СМ-грамматик как форме представления *n*-магазинных автоматов И.В.Вельбицкий со своими учениками основал Р-технология программирования. Задействовавшая графический способ спецификации программ Р-схемами, нормированными в стандартах ГОСТ 19.005-85 и ISO 8631Н, Р-технология поддержана семейством CASE-систем на распространенных компьютерных платформах в Украине и за ее пределами, в частности в России,

США, Франции, Германии, Англии и Чехии. Реализованную по Р-технологии во Франции автоматизированную банковскую систему считали одной из лучших по критериям “компактность /отложенность /цена” и свыше пятнадцати лет эффективно эксплуатировали банки Франции, Люксембурга, Италии, Испании и ряда африканских стран. Технологический Р-комплекс производства программ, функционировавший в реальном времени, поддерживал весь цикл изготовления бортовых систем управления — от проектирования алгоритмов до построения программ и предполетных испытаний моделей ракетно-космических систем. Этот комплекс промышленно эксплуатировался в течение 25 лет и использовался практически для всех спроектированных в Украине и задействованных на боевом дежурстве ракетно-космических систем и систем управления атомными подводными лодками. В составе Р-комплекса продуктивно работал тестово-тренировочный стенд электронных (виртуальных) пусков ракет для всесторонней проверки их систем управления в наземных условиях. А Открытый университет Англии использовал Р-технология в качестве эталона для сравнения визуальных технологий, реализованных в мире в конце XX века.

В 52 организациях СССР, в том числе в Москве, Ленинграде, Минске, Ереване, широкое распространение имела система сборочного программирования АПРОП как “фабрика программ”, конструируемых как на промышленном конвейере из объектов банка разноязыковых модулей. Основу сборочного программирования составили межязыковой и межмодульный интерфейс взаимодействия повторно используемых компонентов, язык описания интерфейса и конструирования отдельных модулей. Имея такой задел в ТП, возглавляемый Е.М.Лаврищевой коллектив легко преодолел барьер начала 1990-х годов, ознаменовавшийся резкой сменой компью-

терных платформ и общего фона проектирования программного продукта открытых систем.

Согласно концепции доказательного программирования Г.Е.Цейтлин формализовал модели и конструкции, удобные для построения строго обоснованных программ с разного рода свойствами, и на основе систем алгоритмических алгебр (САА) В.М.Глушкова создал алгебро-грамматический (АГ) аппарат синтеза и трансформации программ. Посвященная ему монография “Алгебра. Языки. Программирование” дважды издана в СССР (1974, 1978 гг.) и один раз в Германии (1989 г.). Практическим воплощением этого аппарата стала реализация нескольких версий синтезатора “Мульти-процессист” ФОРТРАН-, Си- или Паскаль-программ для разных компьютерных платформ. С середины 1980-х годов в рамках развития АГ-средств многоуровневого структурного проектирования классов алгоритмов и программ разработаны метод поуровневой верификации и трансформации формализованных моделей классов алгоритмов и программ. Аппарат функциональных спецификаций алгоритмов и программ разработан согласно концепции размеченных структур данных. Построенные грамматики структурного проектирования (ГСП) порождают классы алгоритмов символьной обработки, а аппарат гиперсхем развит для формализации смешанных вычислений и конкретизирующего программирования. Аппарат САА и ГСП наиболее развит в направлении формализации концепции абстрактных типов данных как базиса трехосновных САА, детализации и абстрагирования классов алгоритмов и обрабатываемых структур данных путем свертки и развертки базиса алгебры. Вследствие встраивания аппарата алгебраических преобразований усовершенствован инструментарий “Мульти-процессиста”, в котором операторы и предикаты трактуются как средства доступа к базисным объектам трехосновных САА. Таким образом, разработана кон-

цептуально целостная алгебраическая модель процесса многоуровневого проектирования классов алгоритмов и программ на трехосновных САА. Для исследования селективной силы модели построена алгебра обобщенных граф-схем, изоморфная САА Глушкова.

В направлении разработки АГ-методов представления знаний для конструирования профессиональных систем О.Л.Перевозчикова построила асинхронный недетерминированный распознаватель, способный синтезировать класс продукционных систем с динамическим планированием пошаговых диалоговых вычислений. Для распознавателя разработан метод обоснования корректности при синтезе программных систем из набора компонентов многоразового использования. Сформирован набор таких компонентов для прототипирования профессиональных систем по модели предметной области (МПрО), в который включены три группы компонентов: решатель, функционирующий на МПрО; подсистема объяснений об организации вычислений; функциональные блоки (модули или пакеты, реализующие конкретные алгоритмы решения задач). На этой основе реализованы инструментальные комплексы ДИСУППП и Граф-Процесс, в среде которых обеспечивается до 70% применения накопленных компонентов многоразового использования, что соответствует показателям CASE-систем. Посредством этих комплексов реализован десяток профессиональных систем разного назначения (база знаний вычислительной математики, моделирование стратегической стабильности в Европе, принятие решений во время боя бортовым вычислителем роботизированной зенитно-ракетной батареи, проектирование технологических схем обустройства газоконденсатных месторождений на шельфе Охотского моря и проч.).

К концу 1980-х годов резко упали темпы исследований киевской школы ТП,

ее заметно опередили зарубежные разработчики коммерческих CASE-систем, поскольку к тому времени в СССР финансирование НИР уже утратило плановость и системность вследствие “перестройки”. С ее началом стал последовательно разрушаться не только военно-промышленный, но и вообще промышленный комплекс СССР и, естественно, Украины. Это был закономерный итог управления в целом в стране и в информатике в частности, когда всюду наблюдались два недостатка управления: централизм мышления чиновников союзного уровня и пренебрежение достижениями периферийных (провинциальных) научных коллективов; отождествление государственных интересов с собственными, узко отраслевыми.

В таких условиях выживал не сильнейший, а расположенный ближе к властным структурам, принимавшим управленческие решения о распределении критических ресурсов страны. Яркий пример — решение о развертывании в СССР и СЭВ (Совет экономической взаимопомощи) серий ЭВМ ЕС и СМ по заказу нефтегазовиков, стремившихся максимально повысить в СССР добычу нефти и газа для удовлетворения возрастающих потребностей топливно-энергетического комплекса Западной Европы.

Для решения этой задачи требовались новые мощные суперкомпьютеры и вместо совершенствования отечественных ЭВМ “Днепр”, “Минск”, “Наири” (Ереван) решили забыть о собственных компьютерах и адаптировать американские, “взобравшись на плечи гигантов”. Поэтому достижения разработчиков двух поколений отечественных компьютеров ушли в небытие, а СССР стал “догонять” Америку, истратив по крайней мере 15 лет на освоение фактически устаревшей компьютерной техники. Уничтожили также патентно чистый компьютер АСВТ М-4030, прошедший четыре поколения и даже закупаемый Западной Германией. Со временем на основе этого

компьютера были построены вычислительные комплексы ПС-6000, использованные до конца XX века в управлении ядерными энергоблоками на всех АЭС советского производства. Известны ли еще примеры такого массового долголетия компьютерной техники? Кстати, эта техника не имеет отношения к Чернобыльской катастрофе, поскольку для проведения злосчастного эксперимента этот управляющий комплекс сознательно отключили, чтобы он не мешал своим надрызным воем оповещать о наступлении аварийного состояния.

От уничтожения спасли лишь одну БЭСМ-6, работавшую на космический комплекс СССР. На ее базе разработали сначала скоростной спецпроцессор БЭСМ, а потом майнфрейм “Эльбрус”. Именно на его осовремененной базе весной 2001 года заработал первый российский MIMD-суперкомпьютер кластерного типа.

Другой пример чиновничьей некомпетентности — откровенно пренебрежительное отношение к первому в мире ПК серии МИР. К нему крайне отрицательно относились физики, занимавшие в СССР привилегированное положение. Имея возможность выполнять международные проекты в институтах Дубны и ЦЕРНа, физики-атомщики работали лишь на заграничных мощных майнфреймах и программировали только на языке ФОРТРАН с целью обмена алгоритмами с зарубежными коллегами. Когда в конце 1970-х годов выяснилась потребность в создании именно ПК, физики предложили и последовательно отстаивали идею адаптации немецких Wang’ов с входным языком Бейсик, который по функциональности непристойно даже сравнить с языком “АНАЛИТИК” ПК МИР.

Алгебро-грамматические методы агрегирования знаний об объектах и связях алгоритмизируемой предметной области. Как и для большинства научных школ на постсоветской территории, для киевской школы ТП 1990-е годы стали

годами переосмысления направлений исследований и поиска ориентиров и источников выживания. Оглядываясь на пройденный в независимой Украине путь, можно констатировать, что школа не утратила темпа и накопила определенный задел оригинальных собственных исследований, сопоставимых с достижениями мировой программной индустрии, связанными с развитием объектно-ориентированных (ОО) технологий преимущественно под эгидой консорциума OMG, разработавшего индустриальный стандарт CORBA на компонентные модели ОО-проектирования распределенных приложений клиент-серверной архитектуры. Признанные в качестве основного средства реализации больших проектов прикладных систем ОО-технологии преимущественно не имеют специальных средств проверки корректности и целесообразности того или иного комбинирования объектов, оставляя соблюдение правил их функционирования на совести программистов. В то же время традиционные формальные модели и методы представления знаний о предметной области (приложении) ориентированы в основном на использование процедурной технологии. Таким образом, традиционная задача ТП для создания наиболее современного и продуктивного инструментария формулируется как разработка моделей объектного представления знаний о приложении (прикладной системе).

В 90-е годы школа сконцентрировала усилия на исследовании АГ-методов представления знаний о моделях организации вычислений и дружественного интерфейса пользователя (ИП) при проектировании и разработке программных систем, информационных хранилищ и методов извлечения знаний для них, обучающих систем разной ориентации, например систем профессиональной подготовки инвалидов с целью их социальной реабилитации или системы диагностирования интеллекта

чернобыльских детей с задержкой психического развития.

В развитие АГ-средств многоуровневого структурного проектирования классов алгоритмов и программ разработан метод поуровневой верификации и трансформации формализованных моделей классов алгоритмов и программ. Построены ГСП, порождающие классы алгоритмов символьной обработки, и развит аппарат гиперсхем, формализующих смешанные вычисления и конкретизирующее программирование. В синтезе программ ГСП используются как непроцедурные спецификации, а АГ-спецификации не зависят от выбора целевого языка. Подобно смешанным вычислениям процесс генерации программ по АГ-спецификациям трактуется как формирование схем по обобщенным состояниям, которые наряду с алгоритмической частью программы включают обрабатываемые данные. В отличие от конкретизирующего программирования, в аннотациях которого формулируются свойства синтезируемой программы и трансформации, используемые конкретизацией, АГ-аннотации непосредственно в синтезе не используются, а обеспечивается управление выводом в ГСП (детерминированным, недетерминированным, параллельным).

В работах О.Л.Перевозчиковой и ее учеников существенно развиты АГ-методы агрегирования знаний о предметной области и инструментарий для итеративного построения профессиональных систем методом прототипирования по спецификации МПрО согласно спиральной модели жизненного цикла. Эти АГ-методы интегрируют механизмы концептуального моделирования сущностей и связей предметной области с грамматиками и САА. Представление МПрО базируется на формализованной спецификации приложения реляционно-сетевой СУБД "МикроПоиск" в виде графового прототипа, который объединяет модель данных в виде подсхемы БД, модель интерфейса пользователя

как совокупность визуальных формализмов, связанных с объектами БД, и модель вычислений на БД в виде маршрутной схемы.

Синтез системных связей между моделью данных, моделью вычислений, семантическими визуальными формализмами и логикой ИП допускает одновременно и независимо развивать прототип в нескольких направлениях без нарушения целостности программного продукта. К тому же согласно парадигме логического программирования реализован инструмент для получения по МПрО первого прототипа генерируемой системы, а потом ее усовершенствования как реинжиниринга по модифицированным спецификациям отдельных сегментов МПрО. Постепенно с уточнением МПрО повышается способность системы решать задачи своей предметной области, и таким образом формируется интеллектуальная среда разработки прикладных систем.

Понятие самих reuse-компонентов повторного использования категориально расширилось до визуальных формализмов, ознаменовавших переход к ОО-представлению ИП. Независимо от вызываемой программы самонастраивающийся прикладной визуальный модуль обрабатывает символы, организуя отображение, селекцию и редактирование взаимосвязанной информации. Тут под символами понимают любые конструкторы ИТ (текстовые символы, видеоформы, пиктограммы, меню, окна, электронные таблицы и др.), составляющие алфавиты обобщенных грамматик и интерпретируемые как объекты. Визуальный формализм актуализирует текущее состояние информационной среды: от БД, с которой взаимодействует прикладная программа, до отдельных сегментов памяти и информационных структур. Этот формализм строится так, что видеоизображение всегда адекватно отражает актуальное содержание БД.

В разработанную В.Г.Тульчинским объектную АГ-модель ИП для полиэк-

ранного взаимодействия введены грамматики объектного представления ИП и многоосновная алгебра Глушкова на множестве этих грамматик, корректных в отношении стандартов ИП, разработан алгоритм построения базиса алгебры на основе МПрО. Отталкиваясь от построенного асинхронного недетерминированного распознавателя, введена АГ-модель для структурного синтеза маршрутной схемы с сокращенным числом используемых модулей и анализа ее на частичное и полное применение, оптимальность и др. По сравнению с традиционным моделированием программных систем сетями Петри предложенной АГ-модели присуща большая селективная сила.

Усилиями В.О.Гречко, В.Г.Тульчинского, П.Г.Тульчинского АГ-аппарат представления знаний реализован как инструментарий реляционно-сетевой СУБД “МикроПоиск” с развитой инструментальной функциональностью. В “МикроПоиске” реализованы ER-модель данных, QBE- и SQL-подобные языки запросов, поддержка семантических сетей на единой схеме БД для потребностей конечного пользователя. Специальная метабаза используется как хранилище графовых прототипов приложений. Поэтому АГ-аппарат стал эффективным средством трансформации известных фактов в новое алгоритмическое знание. Опыт его эксплуатации подтвердил повышение в 3—5 раз продуктивности труда и качества сгенерированных программных продуктов, среди которых наиболее значимыми являются:

- ❖ система-оболочка “Фактор” для экспертного анализа и прогноза при обосновании управленческих решений, интегрирующая методы прогнозного графа В.Глушкова и анализа иерархий Т.Саати; используется как обучающее пособие по курсам менеджмента на экономических факультетах ряда киевских вузов;
- ❖ комплекс “ГеоПоиск” для оперативной интерпретации результатов гео-

- физических исследований нефтегазовых скважин (сертификат ПА № 3904 от 01.03.2001; см. описание по адресу <http://www.geopoisk.com>);
- ❖ система “ТопоСвязь” для расчетов цифровых сетей электросвязи и коммутируемой междугородней телефонной сети;
 - ❖ диагностический комплекс АСММО массовых медицинских обследований населения, использующий базу знаний по заболеваниям сердечно-сосудистой системы человека;
 - ❖ система-оболочка РТС-Метан для проектирования экологобезопасных схем утилизации угольного метана; использована для опытного проектирования на известной донецкой шахте имени А.Засядько, для Томашевских купольных структур под Лисичанском, Коховских куполов и шахты Западно-Донбасская № 6-42 Павлоградского района Днепропетровской области, шахты Горская государственного холдинга “Первомайскуголь” Луганской обл.;
 - ❖ пакеты автоматического прогнозирования Forecasting ToolPack и Predictor, способные по виду до десяти связанных эконометрических рядов построить четыре прогноза, выбрав наиболее целесообразные из 26 методов прогнозирования, и охарактеризовать для конечного пользователя показатели каждого из четырех построенных прогнозов;
 - ❖ оболочка Magallanes для качественного и количественного рентгено-спектрального анализа.

Хотя названные системы различаются применением механизмов АГ-аппарата, все они имеют значительные по объему МПрО и интенсивно совершенствуются применительно к потребностям обработки неточной и расплывчатой информации. Эта задача тесно связана с реализацией пояснений и рекомендаций в организации ИП, т.е. с разработкой специальных визуальных формализмов для фильтрации и агрегирова-

ния входных данных, по которым устанавливается корректность (частичная или в целом) постановки решаемой задачи. Так, с целью генерации системой АСММО медицинского заключения по результатам компьютерного опроса пациентов с помощью именно визуального формализма устраняется неточность, расплывчатость, противоречивость и неполнота сведений о состоянии здоровья пациентов. Аналогично в системе “Фактор” реализован визуальный формализм для устранения причин возникновения неточности и расплывчатости экспертных суждений путем исследования согласованности элементов матриц мультипликативных парных сравнений. Моделируя поведение экспертов, удалось повысить уровень интеллектуальности ИП и сократить время работы высококвалифицированных специалистов. В 2005 г. на основе построенной онтологической модели метода анализа иерархий модифицирован ИП до полнофункционального Web-сайта.

Эти системы подтвердили качество АГ-спецификации МПрО и встроенного набора визуальных формализмов продолжительностью успешной эксплуатации и заложенным потенциалом реинжиниринга. Так, в среде РТС-Метан удалось в десятки раз сократить срок проектирования и найти наилучшие варианты расположения мощностей по добыче газа, типов используемого технологического оборудования и экологически безопасных режимов его эксплуатации, автоматизировать все экономические расчеты, связанные с эксплуатацией объектов утилизации угольного метана. Этим подтверждены безусловная перспективность системы РТС-Метан и гарантированная окупаемость затрат на ее внедрение.

Комплекс “ГеоПоиск” способен функционировать в единой среде с другими программными продуктами, организуя их взаимодействие и развитие всей технологической цепочки по спиральной модели жизненного цикла автоматизи-

рованного рабочего места геофизика. Как полноценный инструмент “ГеоПоиск” предоставляет сбалансированный набор встроенных утилит, в том числе географические карты и картопостроение (аппроксимация геофизических поверхностей и трехмерная интерпретация), инклинометрию; формирование текстовых документов, планшетов каротажной и другой геофизической информации, корреляционных схем и разрезов; двумерную интерпретацию, кроссплот и статистический анализ; поиск закономерностей, автоматическое создание формул по образцам исследований, в том числе для автоматической идентификации литологии и насыщения. В Украине “ГеоПоиск” используют полевые экспедиции не только для нефтегазоразведки Донецко-Днепровской впадины и близлежащих территорий Азово-Черноморского шельфа, но и для изучения рудных месторождений, а в России — все центры моделирования промышленных нефтедобывающих компаний для Ямало-Ненецкой группы месторождений Западной Сибири. Кроме Украины и России, сегодня комплекс “ГеоПоиск” эксплуатируется еще в 6 странах.

Для интеллектуализации ИП систем Forecasting ToolPack, Predictor, Magallanes разработаны уникальные методы оперативного извлечения и представления упрятанной и неявной информации в накопленных базах и хранилищах данных или во внешней информационной среде, связанные с технологией интеллектуальных (рациональных) агентов:

- ❖ с целью автоматического анализа зависимостей в так называемых больших наборах (БН) пространственно-временных данных реализован метод активного обучения моделей БН и критерий оценки их адекватности, учитывающий оценки отклонений, автокорреляции и сложности модели. Вследствие изучения влияния эффектов, присущих обучению моделей БН данных, и нестабильностей порождающего процесса на

границе ошибок моделей классификации и установления регрессии уточнены оценки риска Вапника—Червоненкиса и установлена связь с границами Хельмбольда и Лонга. Метод задействован в ИП для автоматического прогнозирования временных рядов с полностью либо частично упорядоченным пространством версий;

- ❖ в развитие распределенной разреженной памяти (SDM) Канервы предложена модификация HSDM, ориентированная на упреждающее формирование контекстных рекомендаций и подсказок в интеллектуальном ИП. На основе модификации NSDM, ориентированной на многокритериальное прогнозирование числовых рядов, разработан алгоритм (программный агент) автоматической коррекции числовых прогнозов с учетом подобия образцов для разных сеансов прогнозирования;
- ❖ разработана PAM-модель формирования агентного диалога на основе расписаний, нацеленная на унификацию агентного взаимодействия, и получены расписания автоматического накопления образцов в системе Magallanes.

За результаты по ТП, внедренные в системах Forecasting ToolPack, Predictor, Magallanes, в 2001 г. кандидаты физ.-мат. наук С.В.Зайцева, И.Н.Пшенковская, В.Г.Тульчинский, О.И.Шаров отмечены первой учрежденной Президентом Украины Премией для молодых ученых, которую вручают одновременно с Государственными премиями Украины в области науки и техники. А в 2005 г. в связи с разработкой комплекса “ГеоПоиск” такую премию получили кандидаты физ.-мат. наук А.В.Коломиец, П.Г.Тульчинский и А.В.Харченко.

За последние 25 лет исследования по представлению знаний развились от формализации синтаксиса и семантики языков программирования до извлечения знаний из опытных данных и ны-

нешних агентных технологий. Основываясь на накопленном теоретическом базисе и наращивая темпы познания в информатике, эти исследования продолжают определять современное направление информатики по формализованной поддержке процессов интеллектуализации информационных технологий (ИТ) и нарастающей распределенности информационных ресурсов. Как показал опыт, в ТП развитие прикладных математических моделей тесно связано с практическими достижениями, с рождением, развитием и изменением концепций и парадигм взаимодействия человек—компьютер, характеризующегося в настоящее время отказом от жесткой централизации управления данными и процессами, которая практически невозможна в условиях компьютерных сетей, как глобальных, так и сетей масштаба предприятия (Интернета и интранетов).

Компьютерное обучение. В условиях информатизации общества интерес представляет наделение реализованных программных систем средствами подготовки пользователей. Сформулированную в конце 1960-х годов В.М.Глушковым и Е.Л.Ющенко задачу программированного обучения на компьютерах развил д-р техн. наук, проф. А.М.Довгялло и вместе со своими учениками воплотил в ряде учебно-тренирующих систем. На обобщении опыта разработки и внедрения этих систем потом возникли первые системы популярного в наши дни дистанционного обучения.

В обучающих системах свойство персонализации превращается в способность компьютера адаптироваться к изменяющемуся объему знаний пользователей. Уже в конце 1960-х годов Е.Л.Ющенко с А.М.Довгялло классифицировали три группы обучающихся систем:

1. Обучающие системы в ходе двустороннего обмена вопросами и ответами обеспечивают усвоение материала, выдаваемого компьютером отдельными порциями, как это реализовано в системе ПЕДАГОГ для обучения КОБОЛУ.

2. Обучающе-решающие системы обеспечивают обучение пользователя процедурному языку в ходе решения задач. Пример — система ДИПРОФОР для диалогового решения задач на ФОРТРАНе.

3. Решающе-тренирующие системы предназначены для неподготовленных пользователей, применявших проблемно-ориентированные языки для накопления определенной суммы знаний и навыков. Так, система ДИСКАОД допускала конструировать алгоритмы решения задач обработки экономической информации и получать соответствующие программы на проблемно-ориентированном языке ЯОД-75.

Фактически обучающие системы 1-го поколения были направлены на подготовку не конечных пользователей в современной терминологии, а программистов-разработчиков, поскольку общение с компьютером тогда воспринимали лишь как программирование на каком-либо языке. Потребовались десятилетия, чтобы расширить представление о системах компьютерного обучения. Однако эти первые системы стали уникальным полигоном изучения алгоритмов языкового процессирования, схема которого рассматривалась не как прочная связка переходов от сканирования до парсера, затем к контекстному анализу и дальше к генератору кода. Для организации обучения нужно было поддерживать варианты отката, а это существенно расширяло компиляционные схемы обработки программ.

Для подготовки программистских кадров в 1970-е годы Е.Л.Ющенко основала серию отечественных учебников; несколькими тиражами были изданы (1973—1976 гг.) и переизданы (1979—1986 г.) три учебника по языкам АЛГОЛ-60, ФОРТРАН, КОБОЛ. Благодаря переводу знаменитого аналитического обзора по базам данных, проведенного американским комитетом CODASYL, с энциклопедической полнотой специалистов в СССР впервые

проинформировали об этой важной области информатики (1975 г.). Нельзя не упомянуть о вкладе Е.Л. Ющенко в создание “Энциклопедии кибернетики” (1974 г.) и двух изданий “Словаря по кибернетике” (1979 и 1989 гг.).

Стандартизация ИТ. Уже в 1970-е годы остро встала проблема компьютерной стандартизации не только техники и элементной базы, но и первых интерфейсов — языков программирования, поскольку с появлением языков число их версий быстро приблизилось к 1000 и для каждого нового компьютера разрабатывался свой язык программирования. Тогда же выяснилось, что для стандартизации таких объектов, как программные продукты и операционные среды, необходимы принципиально новые методологии, способные через систематизацию формы (т.е. лишь через интерфейсы) не только повлиять на содержание предмета стандартизации, но и качественно изменить его. В этом состоит новообразующая роль стандартизации как научного познания в сфере информатики, компьютерной техники и телекоммуникации.

Для решения проблемы мобильности программ с начала 1970-х годов одновременно с международным сообществом Е.Л.Ющенко активно занялась стандартизацией языков программирования. В 1977 г. создан ГОСТ языка КОБОЛ, пересмотренный в 1989 г. При непосредственном участии Е.Л. Ющенко и других представителей киевской школы ТП разработаны стандарты СССР по языкам ФОРТРАН-2, АЛГАМС, АЛГОЛ-68, Ада, ФОРТРАН-77, Паскаль, ПЛ/1, Си.

В советское время значение стандартизации ИТ недооценивалось, работы слабо финансировались. Ежегодно героическими усилиями отдельных программистских коллективов и заинтересованных организаций вводились в действие единичные стандарты и потому на момент распада СССР нормативная база стандартизации ИТ оказалась весьма незначительной. Например, до 1990 г. были

разработаны ГОСТы распространенных языков ФОРТРАН-90, КОБОЛ, Паскаль, Си, Ада, ПЛ/1. Однако введены в действие только два из них: КОБОЛ и Ада, а подготовленные проекты стандартов ФОРТРАН-90 (общая разработка России и Беларуси), Си (Россия и Украина) и ПЛ/1 (Россия) так и остались на бумаге из-за отсутствия соглашения о сотрудничестве стран СНГ. Воистину в СНГ совсем не ощущалась потребность в языках ФОРТРАН-90 и Си, а завершёнными проектами стандартов можно было пренебречь.

В независимой Украине уже с 1993 г. начались плановые разработки нормативных документов по стандартизации ИТ. Значительное внимание уделялось формированию украинской терминологии по информатике, компьютерной технике, телекоммуникации и гармонизации с международными ISO/IEC-стандартами. Как член Технического комитета по стандартизации ИТ Е.Л.Ющенко активно включилась в разработку системы, состоящей из 47 терминологических национальных стандартов Украины (ДСТУ) по ИТ, став соавтором девяти ДСТУ по программированию. Эта довольно полная система терминологических ДСТУ обобщила и осовременила категориальную базу, берущую начало от 34 ГОСТов, впервые зафиксировала современные нормы ИТ-терминологической лексики украинского языка и имеет неопределимое значение для информатизации Украины.

В целом с конца 1990-х годов на базе киевской школы ТП работает подкомитет “Языки программирования, их инструментальные среды и системный интерфейс” Технического комитета по стандартизации ИТ, синхронизирующий свою политику с Объединенным комитетом по стандартизации ИТ JTC-1 ISO/IEC. За эти годы разработаны гармонизированные с ISO/IEC-стандартами ДСТУ по:

— независимым от языков типам данных, вызову процедур и спецификациям сервиса (3 ДСТУ);

— процедурам регистрации культурных элементов локализации ИТ и правилам интернационализации ИТ (6 ДСТУ);

— мобильным общим инструментальным средам, POSIX-совместимым средам открытых систем и их администрированию согласно де-факто стандарту CORBA (3 ДСТУ);

— обмену электронными данными для администрирования, электронной коммерции и на транспорте (EDIFACT) и переводу сообщений на EDIFACT-языке в XML-документы (7 ДСТУ).

Несмотря на декларативный характер ДСТУ, введение их в действие не только влияет на становление норм и правил организации компьютерных вычислений, но и способствует повышению качества применения разнообразных ИТ в составе корпоративных информационных систем (КИС).

Напомним, что в конце второго тысячелетия весь мир уже отвлекался на потенциальные негативные последствия функционирования ИТ. На базе киевской школы ТП разработан комплекс средств предотвращения негативных последствий компьютерного кризиса 2000 года. Программный пакет ИК-У2К, разработанный по заказу Госкомсвязи Украины, внедрен на более чем шести тысячах компьютеров, а разработанная вместе с Институтом эксплуатации АЭС Методика испытаний по проблеме У2К введена в действие Государственным агентством по ядерному регулированию Украины для решения “проблемы 2000 года” на АЭС Украины.

Рассмотрим далее влияние стандартов национально-культурной локализации ИТ и возможности сертификации программной составляющей ИТ.

Интернационализация и национально-культурная локализация ИТ — это наиболее современный механизм изменения интерфейса с компьютером для широкого круга пользователей, когда вводятся правила и нормы создания программного продукта, который по-

том во время инсталляции можно настроить на традиции национально-культурного применения ИТ на конкретной территории распространения компьютеров. Речь идет о национально-украинской локализации ИТ с целью повышения технологического статуса украинского языка на основании стандартизированных (узаконенных) средств, допускающих безболезненно переносить и продуктивно использовать в Украине заграничные программные продукты, созданные в другой национально-культурной среде, и, наоборот, достичь международной мобильности украинских программных продуктов, реализовав экспортный потенциал Украины.

Речь идет не только о кодировании информации согласно алфавитам естественных языков (в нашем случае согласно кириллице), но и о фундаментальных проблемах компьютерного моделирования естественных языков усилиями высококвалифицированных и широко осведомленных специалистов. Локализация ИТ означает идентификацию элементов, на которые влияют отличия в языке, культуре, обычаях, привычках, и применение специальных стандартов для этих элементов, чем обеспечивается мобильность ИТ относительно разных национальных культур. Культурные элементы — это сменный компонент интернационализованной ИТ лишь для обеспечения требований культуры пользователя, а функционирование и базовое кодирование ИТ остаются неизменными.

Сейчас международные стандарты определяют процедуры, которых следует придерживаться при подготовке, публикации и поддержке реестра культурных спецификаций для компьютерного применения. Культурные спецификации регистрируются во Всемирном репозитории, получая уникальные идентификаторы стандартного формата, используемые в среде открытых систем, способных по спецификациям автоматически локализовать программ-

ные продукты. Так, Эстония и Литва в 2000 г. зарегистрировали свои наборы культурных элементов.

Заметим, что наибольший объем интернационализации и национальной локализации приходится на языково-текстовый аспект общения с компьютером. Однако речь идет не о локализации самого языка общения (например о русификации или украинизации ИП), а о глубинном пласте обработки естественно-языковой информации, на котором от учета национально-языковой специфики зависит уровень интерпретации информации, т. е. интеллект компьютера. Культурные элементы касаются:

- ❖ кодирования информации согласно алфавитам естественных языков (в Украине согласно кириллице);
- ❖ поддержки электронного документооборота в государстве, организованного по требованиям национальных стандартов ИТ, гармонизированных с международными. Игнорирование этих требований приведет к необратимым процессам уничтожения данных;
- ❖ поддержки множественных естественных языков в ИП компьютеров. Украинские кибернетики и лингвисты (прежде всего лексикологи) с 1960-х годов работают над созданием компьютерной языковой модели. Но любой прогресс в построении моделей и алгоритмов не даст надлежащего результата без больших компьютерных текстовых корпусов естественных языков как технологической базы разработок ИТ и решения на общегосударственном уровне локализации ИТ и, что не менее важно, распространения информации в Интернете.

Как важная культурологическая проблема национально-украинская локализация ИТ непосредственно связана с обретением украинским языком стратегических позиций в государстве и касается создания Корпуса языка — систематизированного, структурированного, про-

граммно обработанного собрания текстов украинского языка во всех вариантах и формах его существования как средства представления украинского языка, культуры, традиций в мировом информационном пространстве. Аналогичные работы в США и Франции начались с 1960-х годов; для славянских языков (в Польше, Чехии, Словении, России, к сожалению, не в Украине) — в 1990-е годы и нацелены на внедрение национальных профилей компьютерной поддержки естественного языка или множества языков. Вместе с упорядоченным правописанием, адаптированным для компьютерного применения, эти профили узаконивают нормы интерфейса (пользователя, отдельных систем и сетей) согласно набору стандартов, численностью около полусотни.

Кроме того, Корпус языка следует рассматривать как часть общегосударственного прикладного профиля, регламентирующего наиболее эффективное функционирование открытых систем в информационном пространстве страны. Вспомним, как с начала 1990-х годов в правительственных GOSIP-профилях США и Великобритании учли специфику E-government-систем и тем самым повысили мобильность, масштабируемость, интероперабельность (способность к взаимодействию с другими компонентами) компьютерных систем и сетей. Именно на таких функциональных или вертикальных профилях основана интеграция всех компонентов и ресурсов единого информационного пространства страны.

Охарактеризуем набор национально-украинских культурных элементов в соответствии с номенклатурой ДСТУ SO/IEC 15897:2003.

1. Лексикографический порядок украинского алфавита, де-факто установленный лишь в школьном букваре, в Германии — в “Staatsgrammatik”.

2. Классификация символов украинского алфавита (большие и маленькие буквы, пунктуация).

3. Числовой формат (десятичная запятая, пробел масштабирования между тройками больших чисел, знаки “+” и “-”).

4. Формат денежных единиц.

5. Соглашения о представлении дат и времени:

❖ *названия дней недели и месяцев (запись полностью, с буквенно-цифровым и цифровым сокращением);*

❖ *порядок дней недели согласно ISO 8601;*

❖ *формат времени (универсального по Гринвичу, местного с правилами перехода на летнее или зимнее время и т.п.);*

❖ *формат (“день.месяц.год” либо “день.год”) арифметики дат для кредитно-финансовых операций.*

6. Положительный (утвердительный) и отрицательный ответ.

7. Национальная терминология по ИТ. Имеем 47 действующих ДСТУ по терминологии ИТ, разработанных в 1993—1994 гг. и уже морально устаревших. Пересмотрены ДСТУ для наиболее значимых разделов ИТ-терминологии по обмену данными, их представлению и организации, базам данных, языкам программирования, распределенным приложениям, электронной почте.

8. Национальные или культурные профили стандартов, отсутствующие, как и большинство из нужных стандартов по ИТ.

9. Набор символов украинского алфавита (базовый набор графических символов украинского языка в составе репертуара современной и старославянской кириллицы, Windows-наборы кириллических символов украинского языка, ASCII-наборы кириллических символов однобайтного кодирования украинского языка и т.п.). К сожалению, уровень поддержки кириллицы большинством операционных систем (в частности Windows, Linux) неудовлетворителен. Ввиду применения сразу шести разных схем кодирования кириллицы зафиксировано много неисправимых ошибок представ-

ления и перекодировки украиноязычных текстов во время переписки по электронной почте или информационного поиска в Интернете.

10. Правила упорядочения для информационного поиска в базах украиноязычных текстов (нормативными документами не установлены).

11. Трансформация букв (транскрипция, транслитерация) алфавита. На замену ГОСТ 16876-71 “Правила транслитерации букв кирилловского алфавита буквами латинского алфавита” следует ввести новые правила транслитерации украинского письма буквами русского и латинского алфавитов. Также для Украины следует транслитерировать алфавиты неславянских языков: крымско-татарского, караимского, гагаузского, возможно, армянского.

12. Характеристика употребления букв украинского алфавита.

13. Использование специальных символов (в частности апострофа, диакритического знака ударения в украинском алфавите).

14. Шрифты (глифы или начертание букв) украинской кириллицы — латинизированной и кириллизированной — для потребностей издательского дела. Напомним, что по начертанию старославянской кириллицы однозначно устанавливается временной период документа, и это важно в компьютерном воспроизведении нашего наследия литературных памятников.

15. Правила письма (все символы и знаки пишутся слева направо).

16. Формирование собственных имен и запись географических названий с их адаптацией (например “Елена, Александр” по-русски и “Олена, Олександр” по-украински; “Харьков, Киев” и “Харків, Київ”).

17. Языковая модуляция по правилам грамматики (склонение существительных и прилагательных, спряжение глаголов, количественные, порядковые и сборные числительные).

18. Правила переноса слов.

19. Пунктуация в украиноязычных текстах.

20. Орфография (правописание) украинского языка.

21. Система мер и весов (согласно ISO 1000).

22. Кодирование национальных атрибутов (территория, население, государственные и региональные языки, соседи, индексы и коды, флаги и гербы, национально-этнические и религиозные праздники и т.п.).

23. Правила телефонной нумерации, в частности международной, и областные префиксы.

24. Почтовая адресация (пятизначный ZIP, общеевропейская форма адреса).

25. Идентификация лиц и организаций.

26. Электронная адресация (следует разработать согласно ISO 3166).

27. Нумерация банковских платежных счетов.

28. Раскладка клавиатуры (имеем устаревший ДСТУ 3470-96).

29. Поддержка диалога в текстовых полях переменной длины.

30. Формат деловых бумаг (преимущественно А4 согласно ISO 838 и ISO 216).

31. Типографские соглашения (имеем ряд ГОСТов).

К сожалению, у национально-украинской локализации ИТ много проблем, обусловленных отсутствием большинства ДСТУ по ИТ. Яркий пример пренебрежительного отношения к национально-украинской локализации ИТ — полное игнорирование с мая 2002 г. представленных проектов двух ДСТУ на схемы однобайтного и мультиоктетного кодирования украинского алфавита в составе кириллицы. Несмотря на бесспорную актуальность этих ДСТУ, до 2006 г. продолжались бесцельная переписка и обсуждение окончательной версии стандартов. Легко спрогнозировать, что с распространением корпоративных систем автоматизации делопроизводства на основании законов Украины об электронном доку-

ментообороте и цифровой подписи выяснятся неопровержимые недостатки применения в Украине не одной, а шести схем кодирования кириллических текстов документов. Справятся ли работники канцелярий и секретариатов с потоком электронных документов, закодированных по одной из шести схем?

Чтобы побудить разработчиков к национально-украинской локализации коммерческих программных продуктов, надо иметь законодательную поддержку как основание для обязательного подтверждения соответствия этих продуктов Закону Украины о языках в УССР от 1989 г. с изменениями и дополнениями, определяющему в статье 2 украинский язык как государственный, а в статье 31 — как язык информатики. Кроме введенных в действие с 2006 г. 6 ДСТУ по процедурам регистрации наборов национально-культурных элементов, только разворачиваются работы по другим вопросам локализации ИТ.

Однако появились первые отрядные признаки перелома ситуации. Так, откликнувшись на потребности просвещения, компания “Microsoft-Украина” провела национально-украинскую локализацию Windows-Vista и Office’2003. К этой работе компания привлекла сотрудников ИК и выступила спонсором издания “Англо-українського тлумачного словника з обчислювальної техніки, програмування та Інтернету”, содержащего около 15 тысяч ИТ-терминов (под редакцией О.Л.Перевозчиковой).

Экспертиза ИТ-проектов. С 1990-х годов мировую моду в программировании и разработке программной составляющей ИТ определяет консорциум OMG — общественная организация с добровольным членством более 900 программистских компаний (ни одной с постсоветской территории). По данным OMG, с начала нового тысячелетия в любом программистском проекте доля программирования составляет 5—7% всех работ, остальное — интеграция “готовых решений”. Кто бывал на еже-

годной выставке CeBIT в Ганновере, хорошо знает лозунги ее более чем 20 выставочных павильонов о применении современных “готовых решений” в любой сфере человеческой деятельности: от конкретного индивидуума до общегосударственных задач политики и управления.

С позиций объективного превращения программистского коллектива, исполняющего любой проект, в команду системных интеграторов, преимущественно оценивающих, отбирающих и сочленяющих в единую ИТ рациональные коммерческие продукты, доступные на рынке, существенно изменилось представление об уровне программистской квалификации ИТ-специалистов. Это нашло отражение в “Рекомендациях по преподаванию информатики в университетах (Computing Curricula 2001: Computer Science)”, составленных в США по инициативе IEEE.

На этом фоне бледнеет и даже утрачивает логику энтузиазм нашей властной и научной элиты, стремящейся отрекламировать и провозгласить программистский потенциал Украины перспективным мощным источником экспортных возможностей страны по поставке наукоемкой продукции, в том числе через технопарки и пр. По данным того же OMG, в настоящее время компании развитых стран проявляют интерес только к уникальным разработкам по:

- ❖ моделированию виртуальной реальности (скажем, в компьютерных играх, электронных театральные постановках, симфонических концертах и мюзиклах, картинных галереях, спортивных мероприятиях и т.п.);
- ❖ средствам решения невиданных по сложности задач на суперкомпьютерах в разнообразных сферах приложений;
- ❖ другим ИТ, реализующим уникальные алгоритмы, преимущественно в гуманитарной или социальной сфере.

Вместе с тем в развитых странах процветает колоссальное множество кон-

салтинговых фирм, за большие деньги помогающих компаниям:

- ❖ выявлять тенденции развития ИТ для эффективной поддержки конкурентного стратегического преимущества выпускаемой продукции на мировом рынке;
- ❖ проводить реинжиниринг (осовременивание) КИС и компьютерных сетей;
- ❖ аттестовать и переучивать обслуживающий ИТ-персонал: системных администраторов баз данных, КИС и компьютерных сетей.

В отношении аттестации персонала проблема весьма остра, поскольку от квалификации системных администраторов (сокращенно сисадминов) зависит качество принимаемых управленческих решений по поддержанию конкурентного преимущества в своем секторе рынка и в конечном итоге судьба компании. Большинство компаний предпочитают нанимать на должности сисадминов только сертифицированных ИТ-специалистов, прошедших обучение в престижных университетах и практику в известных программистских компаниях. Такие программистские гранды, как “Microsoft”, “Oracle”, “Lotus Notes”, за большие деньги проводят семинары для обучения и переквалификации программистов или сисадминов и затем выдают им свои сертификаты.

К сожалению, в Украине отсутствуют центры сертификации программистов, как и инфраструктура оценки соответствия программного продукта стандартам качества. А такая потребность в последние годы уже определилась и это ясно по опыту привлечения ИТ-специалистов ИК для разнообразных, иногда совершенно уникальных экспертиз, предмет которых варьируется от научно-технического уровня выполненного программистского проекта до его стоимости. Назовем только крупные экспертизы, проведенные специалистами возглавляемого О.Л.Перевозчиковой отдела за последние 3 года.

1) В 2005 г. на основе законов Украины и подзаконных актов (6 постановлений КМ) ИК доработал нормативную базу сферы услуг электронной цифровой подписи (ЭЦП) с позиций государственного регулирования этой сферы Центральным заверяющим органом (ЦЗО) согласно Директиве Евросоюза № 93 от 1999 г. За 2006 г. ЦЗО вместе с сотрудниками ИК провел аккредитацию 4 центров сертификации ключей (три в Киеве, один в Харькове), которые развернули инфраструктуру услуг ЭЦП, обслуживая усиленные сертификаты ключей. Реально Украина уже имеет современную инфраструктуру криптозащиты электронного документооборота.

2) В июне 2005 г. совместно с сотрудниками СБУ по заданию Кабмина проведена экспертиза КИС Госкомпредпринимательства, того самого “единого окна”, деятельность которого не раз хвалил Президент Украины. Выявлены кричащие ошибки и должностная халатность, факты откровенного манипулирования тендерами на производство работ, доказан не выдерживающий никакой критики низкий “профессионализм” разработчиков этой КИС из Одессы. К счастью, к нашему мнению прислушались и работу КИС значительно улучшили.

3) В мае 2006 г. по заказу Минтранса провели экспертизу технического задания на реинжиниринг 3-го поколения КИС “Система державного моніторингу міжнародних автомобільних перевезень”. Эта масштабная КИС имеет больше 60 узлов корпоративной сети по всему периметру государственной границы и объемную централизованную базу данных.

4) В ноябре 2004 г. по заказу Госфинуслуг дали положительное экспертное заключение на типовую КИС “Система автоматизації технологічних процесів для адміністрування недержавних пенсійних фондів” на соответствие нормативным документам о функционировании ИТ в сфере негосударственных пенсионных фондов.

5) В мае 2004 г. по просьбе Киевского апелляционного суда (суд 2-й инстанции) провели экспертизу соответствия уставных полей деятельности двух компаний — “Софтлайн-Украина” и “Софтлайн-Россия”. Этот типичный конфликт торговых марок возник вследствие колоссальной некомпетентности “суперспецов” из Торговой палаты. Так, термин “software” они перевели как “програмный сток”. Учитывая аналогичный уровень перевода других терминов, объяснить что-либо судье стало практически невозможно, поэтому экспертное заключение сформулировано с использованием английских терминов.

В последние годы резко возросла потребность в оценке качества задействованных КИС и отдельных ИТ, особенно инициативно разработанных на территории Украины и многие годы поддерживаемых сложившимся коллективом разработчиков. Здесь наблюдаем картину безжалостного и недалевого подхода к используемым ИТ со стороны некомпетентных руководителей предприятий и организаций (государственных и частных), а также граничащий с коррупцией протекционизм многих внедренческих ИТ-фирм, готовых на все, дабы перехватить заказ. Смена руководителя почти всегда приводит к непродуманному пересмотру КИС или ИТ и, значит, к разрушению сложившегося коллектива и реальному провалу КИС.

Организация параллельных вычислений на кластерных системах. С конца 1970-х годов перспективным направлением работы киевской школы ТП стала разработка основ теории и программного обеспечения параллельных вычислений. Впервые концепцию макроконвейерных вычислений озвучил в 1972 г. В.М.Глушков в своем докладе на международной конференции в Италии. Тогда специалисты сразу подхватили эту концепцию, объединившую во всем мире исследования в направлении распараллеливания компьютерных вычислений. Однако фи-

зически на одном процессоре тогдашних майнфреймов сложно было достичь существенного повышения производительности вычислений.

Развернутые теоретические исследования соответствующих моделей и структур данных для параллельных макроконвейерных вычислений практически воплотил в мультипроцессорном вычислительном комплексе ЕС 1766 большой коллектив под руководством Ю.В.Капитоновой, А.А.Летичевского, С.Б.Погребинского, В.П.Клименко, И.Н.Молчанова, А.Г.Кухарчука. К началу 1980-х годов разработан язык мультимодульного параллельного программирования МАЯК и вскоре создана базовая система программного обеспечения комплекса. Благодаря теоретическому заделу удалось достичь рекордной для 1980-х годов эффективности решения практических задач ведения сверхбольших баз данных, обработки структур управления, администрирования технологических процессов.

В рамках развития АГ-аппарата Г.Е.Цейтлин и А.И.Халилов со своими учениками классифицировали стратегии символьной обработки, основываясь на метаправилах конструирования алгоритмов (последовательных и параллельных). Эта классификация введена для формализованных переходов от одного семейства спроектированных алгоритмов к другому, например для генерации классов параллельных алгоритмов. Изучены алгоритмы рекурсивной обработки данных, в том числе синтаксические анализаторы языковых процессоров, для которых в сигнатуре трехосновных САА формализованы проверки левого и правого контекста. Исследована проблема тупиков (зацикливаний) и фиктивности (неисполняемых компонентов итерации) при синхронизации последовательных ветвей в параллельных регулярных схемах программ посредством контрольных точек, семафоров ожидания и рандеву. На основе классификации тупиков установлены критерии тупиковости и фик-

тивности, доказана дуальность задач выявления тупиков и фиктивностей.

Идеи теории конечных преобразователей информации В.М.Глушкова развил д-р физ.-мат. наук, проф. А.В.Анисимов для рекурсивных преобразователей, допускающих обобщение на рекурсивно-параллельные процессы обработки текстов и поддержку скоростной рекурсивно-параллельной арифметики сверхбольших чисел в криптографических алгоритмах защиты информации. Предложенная ПАРКС-технология программирования позволила на единой основе построить рекурсивно-параллельные расширения таких базовых языков программирования, как ФОРТРАН, Си, Паскаль, предвосхитив появление компиляторов MPI-программ.

Настоящего параллелизма вычислений на нескольких или многих скоординированно работающих процессорах достигли лишь с применением сетевых технологий. Произошло это в середине 1990-х годов, а промышленное значение приобрело лишь в конце 1990-х — начале 2000-х.

Ввиду объективных обстоятельств развития Украины в 1990-е годы украинские ИТ-специалисты могли лишь с завистью наблюдать за разработкой и внедрением кластерных систем, преимущественно на Западе, а потом в России и Беларуси. Для закупки за рубежом даже единичных образцов высокопроизводительных компьютеров Украина не имела десятков миллионов долларов, законодательство многих стран—производителей такой техники накладывало жесткие ограничения на экспорт передовых технологий в СНГ, а собственных возможностей создавать такую компьютерную технику Украина не имела.

Решение нашли, только изменив подход к архитектуре суперкомпьютеров кластерного типа, обеспечивающих параллельные вычисления на узлах сублокальной компьютерной сети, в каждом из которых используют процессоры массового производства, а узлы связывают в

сеть высокопроизводительными каналами связи. Такую технику начали производить в Украине в обход ограничений, компоненты каналов связи также выпускают крупными сериями, доступными на рынке. А в качестве общесистемного программного обеспечения практически повсеместно используют свободное программное обеспечение с открытым кодом, доступное через Интернет. Эта архитектура, названная “Беовульф”, очень быстро заняла ведущее место в мире, а образцы таких суперкомпьютеров занимают почти весь перечень TOP500.

Итак, появилась возможность построения суперкомпьютеров согласно финансовой способности всех желающих. Вложив сначала некоторые средства в базовую структуру суперкомпьютера, можно многократно увеличивать его мощность, не заменяя, а лишь пополняя его компоненты. Так, в 2004 г. в ИК на протяжении всего полугодия при активном сотрудничестве с фирмой-интегратором “ЮСТАР” (Киев) и при технической поддержке фирмы “Intel” созданы два суперкомпьютера СКИТ (суперкомпьютер для ИТ) кластерного типа с суммарной вычислительной мощностью по западным меркам университетского уровня. Генеральным конструктором и настоящим энтузиастом разработки стал д-р техн. наук проф. В.Н.Коваль, а его заместителем по программному обеспечению — О.Л.Перевозчикова.

Отметим, что за последние пять лет в современных кластерных системах выяснены такие важнейшие задачи организации параллельных вычислений и программирования:

1) выявление естественного параллелизма классов решаемых задач и применение автоматических механизмов его воплощения, чтобы время создания соответствующего программного продукта для кластерной системы не превышало одного или двух месяцев. Наиболее известный механизм автоматического распараллеливания связывают

с фрагментацией матриц. Внимание разработчиков привлекли такие модели и структуры данных, как деревья и графы, фрагментация обработки которых сулит огромный временной выигрыш в решении задач, скажем, перебора вариантов. Другой интересный объект — SQL-подобные базы данных;

2) выявление минимаксного объема задействованного процессорного ресурса кластерной системы для балансировки оптимальной нагрузки ее узлов в течение решения классов задач;

3) существенное улучшение условий решения классов задач за счет разновидностей архитектуры и топологии соединений узлов кластерной системы, чтобы экономить ресурсы, прежде всего временные и информационные (объем задействованной оперативной памяти или долгосрочных хранилищ). Вдобавок спланированное изучение в этом направлении дает возможность снизить скорость морального старения кластерной системы и стоимость перманентного обновления ее аппаратной составляющей.

Можно констатировать, что для кластерных систем нужны лишь интеллектуальные (построенные на знаниях) ИТ, требующие фундаментальных исследований логико-алгоритмического аппарата, продвинутых операционных сред и инструментов разработки программ. По сути для разработки, например MPI-программ, нужна принципиально новая парадигма программирования.

Ныне международное сообщество заинтересованно следит за всеми достижениями отдельных программистских коллективов-лидеров и конкретными результатами продвижения по этим трем направлениям исследований. Имея непосредственные контакты с несколькими лидерами, представленными коллективами преимущественно американских университетов и исследовательских подразделений фирм-разработчиков, ИК интенсивно обменивается опытом и способен регулярно обновлять инструменты для создания программного продукта пе-

редовых ИТ. Тем не менее, это проходит не всегда безвозмездно (вследствие обмена научными публикациями или сведениями о конкретных программных продуктах) и стоит недешево.

В середине 2004 г. одновременно с наладкой комплекса СКИТ в ИК начаты НИР по реализации ряда ИТ, что позволило собрать команды разработчиков ИТ и применить накопленный задел для решения задач сверхбольшого размера. Довольно быстрое продвижение ИК и получение им нетривиальных результатов обусловлены достижениями киевской школы ТП, для которой научные направления фундаментальных исследований всегда совпадали с указанными. За последние годы результаты этих исследований содействовали развитию математических моделей, методов и инструментов решения важных для практики многокритериальных задач недифференцируемой оптимизации как одного из важнейших направлений многолетних научных исследований ИК:

- ❖ охрана окружающей среды, прежде всего ввиду риска техногенного вмешательства;
- ❖ математическое моделирование состояния разных секторов экономики государства;
- ❖ математическое моделирование процессов и явлений, направленных на эффективное использование энергетических ресурсов, замещение импортированных энергоресурсов и повышение энергетической безопасности Украины;
- ❖ защита информации от несанкционированного доступа или злонамеренного вмешательства в разнообразные информационно-телекоммуникационные системы;
- ❖ обороноспособность государства.

Опыт применения комплекса СКИТ свидетельствует о соответствии мировому уровню организации кластерных вычислений и наличии морально не устаревшего инструмента решения сверхсложных задач супербольшого объема

для нужд по крайней мере Национальной академии наук Украины. Так, в 2005 г. ИК разработал исследовательские образцы восьми интеллектуальных ИТ, макетные образцы которых реализовал еще в 2004 г. на первом поколении СКИТов:

- ❖ ИТ анализа установившегося движения жидкости в естественных трехмерных многокомпонентных грунтовых объектах с полным или частичным влагонасыщением. Решает важную практическую задачу рационального природопользования, в частности для уточнения запасов обеспечения г.Киева питьевой водой и в исследованиях водообеспечения и, как следствие, загрязнения подземных вод в приповерхностных пластах;
- ❖ ИТ для оценки и классификации состояния сложной системы на основе нечетких данных и знаний недетерминированной природы для оценивания и классификации насыщенных симптоматикой состояний сложной социально-экономической системы в зависимости от событий или их свидетельств (под руководством акад. НАН Украины И.С.Дейнеки);
- ❖ ИТ оптимального расположения сервисных центров с незначительным числом поставщиков для обслуживания большого количества потребителей (д-р физ.-мат. наук П.С.Кнопов);
- ❖ ИТ глубинной миграции дуплексных волн (на основе трехмерного полноволнового моделирования данных сейсморазведки земной коры). Применяется в геофизических исследованиях нефтегазоконденсатных и рудных залежей, в частности в Донецко-Днепровской впадине, на Азово-Черноморском шельфе и близлежащих территориях (О.Л.Перевозчикова);
- ❖ ИТ анализа последовательностей ДНК и распознавания структурных элементов пространственной конформации белков. Содействует проведению статистического анализа

любых длинных геномов, затем установлению и подтверждению фундаментальных комплементарных соотношений по записям генетической информации (д-р физ.-мат. наук А.М. Гупал);

- ❖ ИТ оптимизации последовательного обслуживания нескольких динамических объектов в ситуации конфликта и неопределенности (задача динамического коммивояжера). Направлена на решение довольно распространенной в управлении проблемы последовательного достижения нескольких целей, даже в неблагоприятных условиях, в частности в социально-экономической сфере, технике, военном деле (чл.-кор. НАН Украины А.А.Чикрий);
- ❖ ИТ автоматической морфологической разметки украиноязычных текстов с применением словарной базы данных в рамках Национального корпуса украинского языка (О.Л.Перевозчикова);
- ❖ ИТ для матрично-векторных операций и решения задач линейной алгебры с целью создания общесистемных программных инструментов с параллельной организацией вычислений (д-р физ.-мат. наук А.Н.Химич).

Даже этот ограниченный перечень интеллектуальных ИТ, прогрессирующих с каждым годом, позволяет разработать новые механизмы и изучить производительность реализованных инструментов организации вычислений на кластерном комплексе СКИТ. И так, сгруппировав исследователей и объединив их усилия, СКИТ стал не только инструментом решения сверхсложных задач супербольшого объема, но и привел к замыканию обратной связи, когда преодоление проблем теории программирования и организации эффективных вычислений существенно содействует развитию самих интеллектуальных ИТ, способных решать все более сложные задачи.

Для получения представления о специфике реализации указанных ИТ трансвычислительной сложности охарактеризуем одну из них. Четвертую ИТ глубинной миграции дуплексных волн реализовал В.Г.Тульчинский в составе упомянутого комплекса “ГеоПоиск”.

Цель разработки — повышение точности и качества обработки сложных моделей за приемлемое время за счет привлечения больших вычислительных ресурсов. Вместе с тем снижаются требования к квалификации геофизика-интерпретатора, упрощается его работа и уменьшается вероятность ошибок. Разработанные параллельные программы трехмерных глубинных миграций в ходе исследовательской эксплуатации доказали свою полезность по уточнению геологической модели месторождения (рис. 1) и вычислительную эффективность (рис. 2).

Сейсмическое исследование — мощнейший поверхностный способ геологического изучения нефтегазовых месторождений. Бурение одной скважины

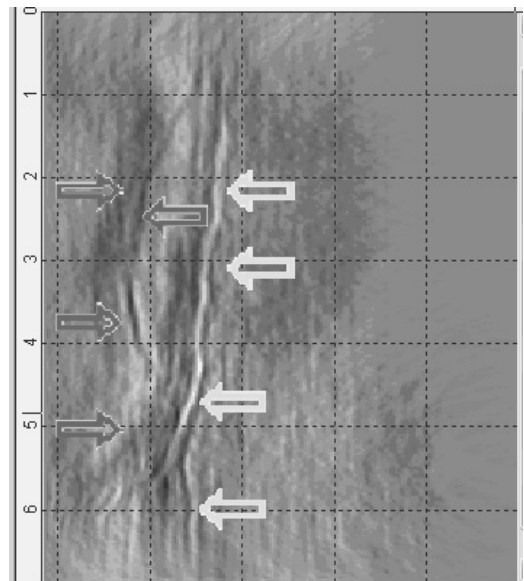


Рис. 1. Горизонтальный срез куба субвертикальных границ, полученного в результате миграции дуплексных волн. Светлые стрелки отмечают известные нарушения, темные — новые тектонические элементы

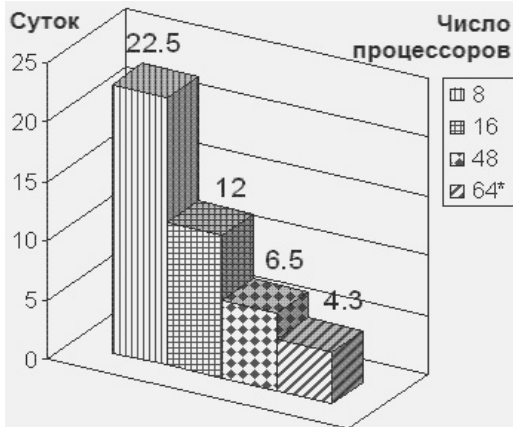


Рис. 2. Сравнение времени миграции трехмерных сейсмических материалов (1,7 млн. трас) в зависимости от числа процессоров кластера

стоит 5—25 млн. грн., что намного превышает стоимость сейсмического исследования полевой экспедицией вдоль многокилометрового маршрута. В условиях постепенного истощения легкодоступных месторождений и соответствующего роста мировых цен приобрела особую актуальность задача разведки и ввода в эксплуатацию нефтегазовых месторождений в районах со сложной геологией, ранее относимых к неперспективным. Рифы, соляные купола и круто наклонные трещины — это типичные примеры сложных геологических структур, ассоциированных с большими месторождениями нефти и газа. В частности, большинство нефтегазовых месторождений Украины имеют сложное геологическое строение.

Традиционные миграционные алгоритмы позволяют получать неискаженные миграционные кубы земной коры лишь в условиях субгоризонтальных прослоек, т.е. при выявлении продуктивных горизонтов в осадочных породах. Разработанная ИТ реализует более сложные современные алгоритмы, пригодные для восстановления с необходимой точностью модели сложноструктурирован-

ных месторождений, в частности миграцию дуплексных волн, разработанную коллективом Украинского государственного геологоразведывательного института (УкрГГРИ) под руководством Н.Я. Мармалевского. Благодаря большому объему трехмерных сейсмограмм, вычислительной сложности алгоритмов и их естественному параллелизму трехмерные миграции целесообразно выполнять на кластерных комплексах.

С 2005 г. начата опытная эксплуатация ИТ в УкрГГРИ, выполнено несколько сервисных проектов. Технология миграции дуплексных волн вызвала значительную заинтересованность в мире. На последних конференциях SEG и EAGE (американского и европейского сообществ геофизиков) ей посвящены отдельные секции и семинары (около 300 участников, в частности представители “Shell”, “Statoil”, “Anadarc”, “Exon Mobil”, BP).

Ныне получен патент США на миграцию дуплексных волн (US 7, 110, 323 B2 от 19.09.2006), завершается оформление соответствующих патентов в Евросоюзе, России, Индии и Китае. Среди геофизических компаний наибольшую заинтересованность проявили несколько канадских фирм (“Tesseral Technologies”, “TetraSeis”) и международная сервисная компания “Kelman Technologies International” с офисами в США, Канаде, Великобритании и Ливии. В 2005 г. суперкомпьютеры СКИТ модернизированы: увеличено количество узлов, чем поднята пиковая производительность до 256 Гфлопс¹, создана единая файловая система кластерного комплекса объемом в 1,7 Тбайт, улучшена инфраструктура электропитания комплекса и его охлаждения, установлена система удаленного мониторинга оборудования комплекса, начата круглосуточная опытная эксплуатация с удаленным управлением. Устоявшаяся работа пользователей через Интернет дает возможность специалистам из других научных

¹ 1 Гфлопс = 1 млрд операций с плавающей запятой в секунду.

учреждений регулярно вести расчеты в предметных областях квантовой химии (Физико-технический институт и Институт сверхнизких температур НАН Украины, Институт радиоэлектроники, г. Харьков), молекулярной биологии (Институт молекулярной биологии и генетики НАН Украины и Институт физиологии КНУ имени Т.Г.Шевченко), обработки спутниковых данных и прогноза погоды (институты Кибернетического центра НАН Украины).

Кластер СКИТ-1 — 48-процессорный 16-узловой кластер на основе микропроцессоров Intel Xeon 2,67 ГГц с разрядностью 32 бита. Кластер СКИТ-2 — 64-процессорный 32-узловой кластер на основе микропроцессоров Intel Itanium 2, 1,4 ГГц с разрядностью 64 бита. Оба кластера управляются своими хост-компьютерами. Коммуникационная среда — InfiniBand, создано единое файловое хранилище емкостью 1,7 Тбайт. Операционная система — Linux. Уже в марте 2005 г. по Интернету поддерживалась связь с комплексом СКИТ удаленных пользователей, решавших задачи прямо с выставочного стенда Всемирной выставки СеВІТ в Ганновере.

В 2005—2006 гг. в ИК разработаны также две ИТ экономического блока для оперативного оценивания динамики макроэкономических показателей экономики Украины и решения задач реструктуризации в отдельных секторах экономики Украины, дискретных оптимизационных задач трансвычислительной сложности для интеллектуального анализа данных и сетевых задач связи.

За 2005—2006 гг. в ИК, кроме кластерного комплекса СКИТ, совместно с киевским НПО “Электронмаш” создан опытный образец рабочей станции ИНПАРКОМ-16. Этот 16-узловой кластер на основе микропроцессоров Intel Xeon 3,2 ГГц с разрядностью 64 бита и дисковой памятью 72 Гб имеет два хост-компьютера (Xeon 3,2 GHz, 8 GB оперативной памяти и 72 GB дисковой). Коммуникационная среда — Gigabit

Ethernet (топологии звезда и гиперкуб), InfiniBand. Операционные системы — Linux и Windows.

Отметим, что под руководством проф. И.Н.Молчанова и д-ра физ.-мат. наук А.Н.Химича для комплекса СКИТ и рабочей станции ИНПАРКОМ-16 разработан интеллектуальный инструмент исследования и решения задач вычислительной математики с приближенно заданными исходными данными, обеспечивающий как автоматический режим полного исследования и решения задач, так и решение задач заранее выбранным алгоритмом из библиотеки geuse-компонентов. В разработанном инструменте воплощены и существенно развиты достижения по распараллеливанию задач линейной алгебры путем фрагментации матриц, выполненные под руководством И.Н.Молчанова в 1980-е годы.

Библиотека geuse-компонентов обеспечивает решение следующих классов задач: систем линейных алгебраических уравнений, алгебраической проблемы собственных значений, систем нелинейных уравнений и систем обыкновенных дифференциальных уравнений с начальными условиями. Накоплен положительный опыт расчетов с использованием интеллектуального инструмента для задач обтекания НПО “Антонов”, теплопередачи Института электросварки им. Е.О.Патона НАН Украины, прочности строительных конструкций. Так, для расчетов прочности конструкций из библиотеки geuse-компонентов задействованы средства:

- ❖ формирования геометрической модели конструкции и математической модели задачи на основе имеющихся в библиотеке geuse-компонентов;
- ❖ формирования конечно-элементной или конечно-разностной модели задачи;
- ❖ визуализации конечно-элементного покрытия элемента или исследуемой конструкции;
- ❖ формирования в автоматическом режиме дискретной модели задачи и загрузки данных на процессоры

кластера выбранной топологии соединений узлов;

- ❖ вызова МРІ-программ для исследования и решения сформулированных конечно-элементных задач;
- ❖ анализа достоверности полученного конечно-элементного или конечно-разностного решения с их визуализацией.

Заключение. Преодолев полувековой путь исследований, ТП обогатилась собственным развитым формально-алгоритмическим аппаратом, а предмет ее исследований существенно расширился от процедурных языков до методов представления знаний, составляющих искусственный интеллект инструментария разработчиков прикладных систем. Как способ общения с компьютером и средство разработки программ языки трансформировались в алгоритмические системы со сбалансированными наборами конструкторов, срабатывающих в операционной среде с достаточно формализованным и унифицированным поведением, правила которого совместно с языками программирования стандартизируются усилиями интернациональных коллективов.

В начале третьего тысячелетия сформулирована новая задача ТП — разработка методологической основы ИТ, немислимым без распределенной обработки в глобальной телекоммуникационной среде. Ныне наблюдается вовлечение в решение этой задачи с невероятно расширившимся спектром проблем специалистов из других направлений компьютерных наук, ранее занимавшихся методами вычислительной математики и статистики, инженерными проблемами создания компьютерной и телекоммуникационной техники и пр.

Благодарим соратников, учеников и последователей Е.Л.Ющенко за оказание помощи и неподдельный энтузиазм в подборе материалов и подготовке этой публикации, посвященной нашей незабвенной Е.Л.Ющенко. Ее жизненный путь восхищает богатством идей и мастерством ученого, отзывчивостью педагога и неис-

сяжаемым трудолюбием. Е.Л.Ющенко — автор более 200 научных работ, в их числе три десятка монографий и учебных пособий, пять авторских свидетельств на изобретения, десяток национальных стандартов Украины и СССР. Часть из этих работ вышла двумя-тремя тиражами и переведена за границу. За пионерские работы по программированию Екатерине Логвиновне присуждены две Государственных премии Украины и Премия Совета Министров СССР, а теоретические результаты по компьютерной алгебре отмечены премией имени В.М. Глушкова. Украина высоко оценила заслуги Е.Л. Ющенко, наградив ее орденом Княгини Ольги.

Много сил и энергии отдала Екатерина Логвиновна подготовке научной смены. Среди ее учеников 47 кандидатов и 11 докторов наук. Свою исследовательскую и научно-организационную работу Е.Л. Ющенко совмещала с активной общественной работой, будучи членом редколлегии журналов “Кибернетика и системный анализ” и “Проблемы программирования”, бессменным членом нескольких специализированных советов по присуждению ученых степеней и популярным оппонентом огромного числа диссертаций. Именно в этом качестве ее знает и с благодарностью вспоминает не одно поколение программистов, для становления которых как ученых она не жалела времени и сил, помогая отшлифовать формулировки научных результатов.

Дадим возможность И.Н.Вельбицкому поделиться своими дорогими воспоминаниями.

“Екатерина Логвиновна Ющенко — человек-легенда Института кибернетики им. В.М. Глушкова НАН Украины. Как ученого её можно отнести к представителям старой классической школы, для которых истина, поиск истины и методы поиска, их чистота и качество, доказательность результата были превыше всего. Время, в котором она активно работала, было временем быстрой генерации и

частой смены нового, рождения всех фундаментальных идей существующей науки программирования. В то время в нашей науке было много мутного, псевдофундаментального и, как выяснилось, с очень малым периодом полураспада. Екатерина Логвиновна как ученый имела острое чутьё на все новое, которое она выделяла не только своим острым логическим умом и даром гениального предвидения, но и сердцем как женщина, убеждая поверить в это новое окружающих своим личным примером, работая в новом направлении.

Трудоспособности она была необыкновенной. Работа с ней — это школа жизни в науке. До сих пор с благодарностью вспоминаю, как полтора года почти ежедневного общения с ней мне приходилось доказывать новизну и универсальность своего подхода визуального программирования и формального описания синтаксиса и семантики языков с помощью нагруженных по дугам графов. Эмоции и амбиции молодости наткнулись на твёрдое и беспелляционное Учителя: “не понимаю, нет доказательства универсальности”. Борьба с этими тезисами заставила перелопатить огромное количество литературы и задуматься над вопросом, что же такое понимание, смысл, семантика, как сделать новое понятным хотя бы Учителю, уже не говоря о самом себе, объяснить, почему не понятно, и сделать все формально понятным даже машине. Запомнился на всю жизнь и переломный момент в человеческих отношениях, когда по её заданию были описаны синтаксис и семантика пустого языка в новой

нотации в сравнении с известными способами (и это после формального описания таких языков, как Алгол, Паскаль и фрагментов PL/1, Си! Тогда это казалось несправедливым по времени и трудозатратам на эти работы). После этого начался этап эскалации научных исследований и публикаций в одобренном Учителем направлении.

На этом этапе проявилось ещё одно качество Учителя — её способность подачи научного материала в соответствии с требованиями времени, способность отточить и лаконично сформулировать любую мысль, убрав из неё все неточности, двусмысленности понимания и толкования написанного.

Редактором и Учителем она была от Бога. Соавторство публикаций с ней, а их было более 50, тоже большая школа, которая работала и работает в моей жизни до сих пор. Наиболее ярким её проявлением, например, была работа в 1980-х гг. в Международном комитете ISO для стандартизации нашей новой системы визуального программирования. Обычно новый стандарт готовится в ISO неторопливо в течение 4—5 лет. Нам же удалось за одну 10-дневную командировку включить свою нотацию в новый стандарт ISO. Ещё три командировки за счет ISO были потрачены на то, чтобы доказать, что наши предложения лучше японских, которые рассматривались Комитетом в течение 4 предыдущих лет.

Спасибо большое за всё, дорогая Екатерина Логвиновна! Пусть земля будет Вам пухом, а память учеников и потомков вечной...”

Получено 12.09.2007

О.Л. Перевозчикова

Школа теорії програмування К.Л. Ющенко

Огляд присвячено історії виникнення і сьогодинньому дню школи теоретичного програмування, що до початку нового тисячоліття (на протязі 40 років) працювала під керівництвом члена-кореспондента НАН України К.Л.Ющенко на базі створеного нею в 1957 році відділу автоматизації програмування Інституту кібернетики імені В.М.Глушкова НАН України. З позицій сьогодення зроблено спробу оцінити роботи К.Л.Ющенко та її найближчих соратників та особливо учнів — колишніх аспірантів: докторів фіз.-мат. наук, професорів І.В.Вельбицького, О.М.Лаврищевої та О.І.Халілова, доктора техн. наук, професора Г.Є.Цейтліна, чл.-кор. НАН України, д-ра фіз.-мат. наук, професора О.Л.Перевозчикової.