

Статьей Г. Р. Громова журнал завершит публикацию материалов «круглого стола», который проходил в Большом зале Политехнического музея 4 октября 1985 г. (см. «МП», 1986, № 2, с. 92—93).

УДК 681.322.1-181.4+681.338.45

Г. Р. Громов

АВТОФОРМАЛИЗАЦИЯ ПРОФЕССИОНАЛЬНЫХ ЗНАНИЙ

Введение

Темп развития технологической цивилизации определяется в значительной степени темпом накопления профессиональных знаний. В свою очередь общая сумма потенциально доступных членам человеческого общества знаний зависит от достигнутого на данном историческом этапе уровня эффективности процесса «отчуждения» индивидуально генерируемых «частич» знания от автора — их первичного источника и начального носителя; человека, который первым овладел тем или иным новым технологическим приемом, методом, средством и т. д.

Истоки информационной технологии

На ранних этапах развития цивилизации профессиональные навыки передавались в основном личным примером исполнения производственных действий (новых приемов охоты, обработки шкуры, костей животных и т. д.). Рациональные способы организации коллективных действий, синхронизация производственных усилий закреплялись для передачи из поколения в поколение ритуальными танцами, обрядовыми песнями, устными преданиями и т. д.

«Помехоустойчивость» социально-исторического канала передачи профессиональных знаний заметно возросла с открытием человеком элементов технологии длительного хранения на материальном носителе отдельных, наиболее характерных зрительных образов, связанных с накопленными знаниями. Произошло это по историческим чертам совсем недавно — каких-нибудь двадцать—тридцать тысяч лет назад. Именно тогда появились первые наскальные рисунки. Возраст человеческой цивилизации составлял к этому времени сотни тысячелетий.

Шесть тысяч лет назад технология регистрации на материальном носителе символики кодированной информации о накопленных знаниях достигла того порогового уровня, с которого ведут отсчет эры письменности. Таким образом, за каких-нибудь двадцать тысяч лет человеком был пройден путь от наскальных рисунков до первых глиняных табличек с текстами. Это был путь поиска все более совершенных способов кодирования и расшифровки фиксируемых для длительного хранения на

материальном носителе элементов знаний.

Начатый тогда процесс совершенствования носителей информации и инструментов для ее регистрации продолжается до сих пор: камень, кость, дерево, глина, папирус, шелк, бумага, люминофор, магнитные и оптические носители, кремний, цилиндрические магнитные домены и т. д.

Однако, накапливаемые в виде отдельных записей или книг профессиональные знания не могли непосредственно влиять на производственный процесс. Чтобы получить шанс «прорасти» новым знанием или повлиять на характер выполняемого другими людьми трудового процесса, книга должна была попасть «на благодатную почву» — на нее должен был «наткнуться» читатель, который в силу редчайшего стечения обстоятельств оказался уже подготовлен собственной биографией к свершению «таинства зачатия» новой идеи именно в данной профессиональной области знаний. Иными словами, только в том до невероятного редком случае, когда у автора книги и одного из немногочисленных читателей дорогого рукописного фолианта оказывался «резонанс» конструктивных идей, книга могла способствовать акту рождения нового знания. Понятно, какое воздействие на темпы развития технологической цивилизации должно

было зафиксированной на материальном носителе информации о новых знаниях значительно повышалась вероятность события, что хотя бы одно «семя знания» попадет на благодатную почву», прорвет и в свою очередь даст «массовым тиражом» обогащенное новым знанием свое собственное «послание в будущее».

Стимулируемое книгопечатанием развитие наук ускоряло темпы накопления систематизированных по отраслям знаний. Эти знания теперь можно было быстро тиражировать и они становились доступными для многих нередко далеко удаленных друг от друга территориально и во времени участников внутриотраслевого трудового процесса. Например, при создании паровой машины основные технические решения были получены врачом Д. Папеном (1690 г.), шихмейстером Кольваго-Вознесенских заводов И. Ползуновым (1763 г.), лаборантом университета в Глазго Дж. Уаттом (1769 г.). «Паровая машина была первым действительно интернациональным изобретением...» — отмечал Энгельс.

За три столетия после изобретения в 1445 г. печатного станка сказалось возможным накопить ту «критическую массу» социально доступных знаний, при которой начался лавинообразный процесс развития промышленной революции.

«Наука — это та часть наших знаний, которую мы сумели понять настолько хорошо, что можем обучить этому ЭВМ. Там, где мы еще не достигли такого уровня понимания, речь пока идет лишь о профессиональном искусстве. Формальная запись алгоритма или программы ЭВМ, по-существу, позволяет нам выполнить весьма полезный тест глубины наших знаний, так как переход от искусства к науке просто означает, что мы поняли, наконец, как автоматизировать данную предметную область».

Д. Кнут

было в этих условиях оказать изобретение печатного станка — машины для тиражирования зафиксированных на материальном носителе знаний.

Книгопечатание — первая информационная революция

Книгопечатание выполняло для роста накапливаемых человечеством профессиональных знаний ту же роль, какую играет, например, для растений рассеяние семян. Массовое тиражирование для последующего «рассеяния» на больших простран-

ствах овеществленные через трудовой процесс в станки, машины, новые технологические процессы и другие организационно-технические новшества, становились источником новых идей и плодотворных научных направлений. Регенеративный цикл: знания — общественное производство — знания, оказался замкнут, и спираль технологической цивилизации начала раскручиваться с нарастающей скоростью. Печатный станок сыграл при запуске этого процесса роль информационного ключа, резко повысив пропускную способность социального канала обмена знаниями.

Математический метод

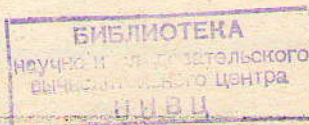
С разлуки с потребностями функционирования по основным элементам отчужденности степени «таинственности» нем все нового на производстве время на даментальность элемент профессора сегодня шим по

Источники видимо, нам, когда профессиональные революции отказывались роль на переход к цивилизации ственно первые «звездочка» клинчатом ростом требности методом ми и «сектор» начали «донауча» современны» для

Напри статично ли у ми счелетии нем вл спация в буквальн ски» в е номическ сосредот этой за среднен ны пер «тайн в храните

Успеш формализ ний пов парусни ментами таблица мулами любого ходимок лет вос чета».

Успех чуждени от их «самого даясь в ностью»



Математика — базовая методология индустрии знаний

С развитием промышленной революции становилась все более острой потребность в создании регулярно функционирующего научного аппарата по основным областям профессиональных знаний, чтобы процесс их отчуждения от автора все в меньшей степени оставался внутрицеховым «тайнством», а приобретал со временем все более характерные черты одного из рутинных этапов типового производственного процесса. В это время начали закладываться те фундаментальные понятия и основные элементы технологии формализации профессиональных знаний, которые сегодня иногда объединяют более общим понятием «индустрия знаний».

Истоки этого процесса восходят, видимо, к тем незапамтным временам, когда жрецы — первые профессиональные хранители общинных сокровищ знаний, начали постепенно отказываться от претензии на контроль над всей необъятной «магией» и переходить к индивидуальной специализации по областям «преимущественных интересов». Так возникли первые «специалисты»: астрономы — «звездочеты», психотерапевты — «заключатели болезней» и т. д. Вместе с ростом социально-экономической потребности в более конкретном предметном знании из этих тысячелетиями изолированно развивавшихся «секторов» когда-то единой «магии» начали формироваться исторически «донаучные», а затем и некоторые современные научные и «околонаучные» дисциплины.

Например, жрецы-«звездочеты» достаточно высокой квалификации были у многих народов земли еще тысячелетия назад. Однако, с наступлением эпохи мореплавания «дефицит» специалистов этого профиля стал в буквальном смысле «катастрофически» очевиден. Острая социально-экономическая потребность вынудила сосредоточить на решении именно этой задачи «лучшие научные силы» средневековья, и скоро стали заметны первые успехи в отчуждении «тайн неба» от их «богоизбранных» хранителей.

Успешно развивавшийся процесс формализации астрономических знаний позволил оснастить океанские парусники навигационными инструментами, а книги с астрономическими таблицами, схемами и точными формулами позволяли практически из любого грамотного отрока при необходимом прилежании за несколько лет воспитать корабельного «звездочета».

Успех (или провал) попытки отчуждения профессиональных знаний от их «богоизбранных» носителей до самого последнего времени определялся возможностью (или невозможностью) их формализации математи-

ческими методами. Области профессиональных знаний, которые оказались доступны для такого подхода, получили название «точных наук».

Процесс формализации знаний, как правило, сводился к тому, чтобы попытаться из всего многообразия сведений в избранной области человеческой деятельности выделить небольшую, но логически определяющую достаточно многое зону доступного

низированному «критерию истинности». Более того, за последние десятилетия, и особенно в первые годы «эры ЭВМ», относительный вес «точных наук» в общей системе научного знания начал, как принято было считать, возрастать еще более быстрыми темпами.

В начале 50-х годов возник «кибернетический бум», суть которого заключалась в стремительно распространявшейся, главным образом, среди представителей «точного

За три столетия после изобретения в 1445 г. печатного станка оказалось возможным накопить ту «критическую массу» социально доступных знаний, при которой начался лавинообразный процесс развития промышленной революции.

...Печатный станок сыграл при запуске этого процесса роль информационного ключа, резко повысив пропускную способность социального канала обмена знаниями.

математическим методом «формализуемого ядра». Это позволяло в случае успеха создать формально строгую «локальную систему знаний». В результате значительная часть содержательных сообщений об очередном «приращении» знаний в рамках данной предметной области могла быть исчерпывающе изложена на ограниченном формальными правилами языке, лишенном «недостатков разночтения». При этом оказывалось, что текст мог быть либо понятным и логически однозначно истолкованным для всего профессионального сообщества, либо квалифицирован как неправильный. Это означало, кроме прочего, что процесс передачи знаний от автора печатной работы заинтересованному читателю в рамках созданной «локальной системы знаний» оказывался уже не должжданным редким событием, лишь на вероятность которого можно было, и то лишь косвенно, влиять тиражом издания, а практически достоверной рабочей процедурой.

Позитивный результат, достигнутый в формализации любой социально значимой области человеческих знаний, создавал эффект, с которым можно было сравнить лишь библейское чудо прозрения. Достаточно упомянуть известный поэтический комментарий к научному подвигу И. Ньютона:

«Был этот мир глубокой тьмой окутан.

Да будет свет!
И вот явился Ньютон»...

Понятно поэтому, что основным критерием для отделения науки от искусства, ремесла и других форм «донаучной» деятельности был принят уровень их математизации. «Учение о природе будет содержать науку в собственном смысле лишь в той мере, в какой может быть применена в ней математика» — отмечал Иммануил Кант. До самого последнего времени в «учении о природе», казалось, не возникало сколько-нибудь заметных фактов, способных поколебать доверие академической обществу к этому давно уже кано-

нальному «критерию истинности». Более того, за последние десятилетия, и особенно в первые годы «эры ЭВМ», относительный вес «точных наук» в общей системе научного знания начал, как принято было считать, возрастать еще более быстрыми темпами.

Аналогичным «кавалерийским атакам» подверглись в тот период экономика, психология и многие другие ранее малодоступные для традиционной математики области научной деятельности. Видимо, наиболее точно описал атмосферу «математического клондайка» первого десятилетия компьютерной эры Клод Шеннон. В опубликованной тогда научно-публицистической статье «Бандвагон» он прямо предупреждал своих коллег: «Здание нашего несколько искусственно созданного благополучия слишком легко может рухнуть...»

Экономика компьютеризации

В чем же заключалась реальная притягательная сила, так сказать, рациональное зерно «кибернетической эйфории» пятидесяти лет, для многих из тех широко известных своим конструктивными результатами выдающихся ученых, которых она, по крайней мере, в самое первое время, вовлекла в свою орбиту.

Видимо, главным образом это было вызвано тем обстоятельством, что с появлением в середине XX века станков для обработки информации — ЭВМ, впервые в человеческой истории оказался возможным такой способ записи и долговременного хранения ранее формализованных математическими методами профессиональных знаний, при котором эти знания могли непосредственно, без промежуточного воздействия на человека, влиять на режим работы производственного оборудования. Процесс записи ранее формализованных профессиональных знаний в готовой для непосредственного воздействия на машины и механизмы форме получил название — «программирование ЭВМ».

Можно было ожидать, что появление станков для непосредственного

включения в производственный процесс огромной массы накопленных человеческим профессиональным знаниям, вызов резкий, «взрывной» рост производительности труда. Однако, как известно, за первые 30 лет компьютерной эры этого так и не произошло. В США, например, огромные и быстро растущие расходы на вычислительную технику сопровождались в период с начала 50-х до конца 70-х годов неуклонным снижением темпов роста производительности труда. Основная причина наблюдаемого за последние десятилетия снижения темпов роста производительности труда в промышленно-развитых странах — непрерывный отток людей из сферы материального производства в «информационную сферу» народного хозяйства. Рост численности «информационных рабочих» «knowledge workers» вызван постоянным увеличением сложности индустриального общества и, как следствие, объема циркулирующих в нем информационных потоков. Однако, если машины и системы автоматизации в сфере материального производства постоянно совершенствовались и, соответственно, производительность труда там возрастала, то в сфере обработки информации, где трудятся ученые и специалисты, служащие, руководители всех уровней — средства автоматизации проникали до сих пор с большим трудом. Численность людей, занятых в информационной сфере, составляла поэтому к началу 80-х годов в большинстве промышленно развитых стран уже около 50 % от общего числа занятых во всех отраслях народного хозяйства и продолжала быстро расти. Естественно было бы в этих условиях задать очевидный вопрос: чем же объяснить, что за 30 лет эры ЭВМ уровень автоматизации труда людей, работающих в сфере обработки информации, в среднем почти не изменился?

Барьер формализованных знаний

Дело в том, что первые поколения ЭВМ были созданы для решения в основном лишь хорошо поставленных математических задач и, в первую очередь, задач чисто расчетного характера. «Ибо это недостойно рода человеческого, подобно рабам тратить часы на вычисления» — столетиями сокрушались математики. Созданная для решения такого типа задач вычислительная машина стала со временем полезным инструментом и в целом ряде других областей приложений. В основном ЭВМ применялись там, где необходимо было решать корректно поставленные на формальном уровне задачи. Как правило, это были наиболее приоритетные народно-хозяйственные и оборонные задачи из достаточно «продвинутых» по уровню накопленного

задела математических методов областей «точных наук». Успешными оказались и попытки решения на ЭВМ так называемых информационных задач, существо которых обычно заключается в необходимости автоматизировать процесс хранения и быстрой обработки больших объемов хорошо структурированных данных по стандартным алгоритмам: поиск патентной информации или библиографических ссылок, некоторые банковские операции, продажа авиабилетов, и т. д.

Однако все попытки в поисках эффективных областей приложений ЭВМ выйти за пределы ранее накопленного задела формализованных задач и ранее сложившегося в хозяйственном механизме фонда структурированных данных наталкивались на значительные, быстро растущие (по мере углубления понимания предметной области) трудности.

Успех (или провал) попытки отчуждения профессиональных знаний от их «богоизбранных» носителей до самого последнего времени определялся возможностью (или невозможностью) их формализации математическими методами. Области профессиональных знаний, которые оказались доступны для такого подхода, получили название «точных наук».

Как заметил об этом К. Шеннон, «ЭВМ выглядят как ученые холосты. При вычислении длинной цепи арифметических операций ЭВМ очень значительно обгоняют человека. Когда же пытаются приспособить ЭВМ для выполнения неарифметических операций, они оказываются неуклюжими и неприспособленными для такой работы».

Все это вынудило на рубеже 80-х годов несколько более критически оглянуться на пройденный путь. При этом выяснилось, что, как и предсказывал Шеннон, «в здании нашего несколько искусственно созданного благополучия» видны зияющие провалы.

Кроме упомянутого выше примера низведения тысячелетиями охраняемых таинств «жрецов-звездочетов» до уровня широко доступных практическому использованию законов небесной механики, можно было бы привести множество и других столь же убедительных примеров из самых различных областей «точных наук», показывающих, как процесс формализации знаний завершался, в случае успеха, разработкой математически строгой техники отчуждения «внутрицеховых» профессиональных тайн. Однако не менее важно и то, что далеко не все «форты» профессиональных знаний удавалось до сих пор брать «лобовой атакой», используя лишь тот арсенал средств формализации, которым располагала традиционная математика. Например, передача профессиональных знаний в таких важнейших областях современной науки, как медицина, происходит и сегодня во многом теми же сред-

ствами, что и тысячу лет назад. Причем происходит это вовсе не от недостатка внимания сторонников «точных методов» к этой древнейшей из наук. Как заметил академик И. М. Гельфанд, подводя итоги 15 лет работы по медицинской диагностике и прогнозированию возглавляемого им коллектива математиков и врачей, «применение математических методов в медицине, несмотря на относительно длинную историю, все еще находится в начальной стадии. При первых же столкновениях с реальным медицинским материалом стало ясно, что те испытанные общие принципы, с которыми математики подходили к физическим и техническим задачам, в этой новой области плохо применимы. Аналогичное положение дел имеет место, по-видимому, и в других нетрадиционных для применения математики областях».

Развивая мысль, которую И. М. Гельфанд вкладывает в заключительное обобщение, можно было бы отметить, что медицина далеко не единственный «крепкий орешек» для традиционных методов формализации профессиональных знаний. Дело здесь, видимо, в существенно разном уровне сложности задач, которые возникают при необходимости вычислить баллистическую траекторию, оценить ожидаемую температуру газа, или же механизм функционирования живой клетки, целостного человеческого организма, производственно-го предприятия, отрасли, и т. д.

«Совершенно естественно, — отмечал в этом контексте Ж. Адамар, — говорить об уме более интуитивном, когда зона комбинирования идей находится глубоко, и об уме логическом, если эта зона расположена достаточно поверхностно».

Структура фонда профессиональных знаний

Чтобы попытаться оценить, как соотносятся между собой по отношению к общему объему различные «слои» накопленного человечеством профессиональных знаний, давайте предельно в этом зале простоты идей находимся глубоко, и об уме логическом, если эта зона расположена достаточно поверхностно».

Давайте примем за 100 % объем ваших знаний в той области, где вы считаете себя наиболее сильным в профессиональном отношении. А теперь попытайтесь оценить, какую часть из полного объема ваших про-

фессии переда боте. ет вас другим смогла коллег ными минка превыш ваших что вл лишь этом даже разите но-язы 1 % о предс ных с вас ос ных о мулы, ванья гаю, с вующи меньш еще на Всея ловече принци мализа ции на воар, форма ка при на нем форма «пленн послед стью 1 тодова дач. О ки», х известн часть глублин которе деятел рых вать, и потен ции замеч то ичн нередк органи счетаи как эт редине ными л Ины фонда профес быть п жающ основ лежит му об послед недося ло «и этого ваемы ки, при

фессиональных знаний вы могли бы передать ближайшему коллеге по работе, т.е. человеку, который понимает вас в рабочем контакте лучше других? Моя оценка: то, что вы смогли бы объяснить ближайшему коллеге, пользуясь любыми доступными вам средствами общения (речь, мимика, жесты, чертежи и т.д.), не превышает 20...10% общего объема ваших знаний. Теперь предположим, что вы можете общаться с коллегой лишь письменно. Я полагаю, что в этом случае вы сможете передать, даже пользуясь всем богатством выразительных возможностей естественно-языковых текстов, уже не более 1% объема ваших знаний. Наконец, предположим, что из всех возможных способов передачи знаний для вас оставлено только язык формальных описаний: математические формулы, известные языки программирования и т.д. В этом случае, я полагаю, основная часть здесь присутствующих смогли бы передать еще меньшую часть своих знаний, видимо, еще на несколько порядков меньше...

Вся область профессиональной человеческой деятельности, которая принципиально поддается пока формализации, а, значит, и автоматизации на базе ЭВМ, — это, образно говоря, тонкая поверхностная пленка формализованных знаний, лишь слегка прикрывающая поверхность океана накопленного человечеством неформального знания. Именно эта «пленка» и оставалась до самого последнего времени доступной областью для приложения машинных методов решения интеллектуальных задач. Отношение толщины этой «пленки», характеризующей доступную известным методом формализации часть человеческих знаний, к общей глубине профессиональных знаний, которыми оперируют в повседневной деятельности работники, труд которых предполагается автоматизировать, и является сегодня показателем потенциально достижимой эффективности внедрения ЭВМ. Попытки не замечать этих ограничений, или просто игнорировать их волевым образом нередко приводили к значительным организационно-экономическим просчетам, крупномасштабным потерям, как это, например, происходило в середине 70-х годов с печально известными АСУ.

Иными словами, общая структура фонда накопленных человечеством профессиональных знаний может быть представлена в виде быстро сужающейся по высоте пирамиды. В основании этой «пирамиды знаний» лежит самый значительный по общему объему слой, который до самого последнего времени был практически недостижим для какого бы то ни было «внешнего доступа». Элементы этого слоя — индивидуально накапливаемые «мастерами» знания и навыки, принципиально неотчуждаемые от

...с появлением в середине XX в. станков для обработки информации — ЭВМ, впервые в человеческой истории оказался возможным такой способ записи и долговременного хранения ранее формализованных математическими методами профессиональных знаний, при котором эти знания могли непосредственно, без промежуточного воздействия на человека, влиять на режим работы производственного оборудования.

их авторов традиционными методами формализации: «Могу сделать, но не знаю, как это объяснить».

Расположенный выше и, соответственно, значительно меньший по относительному объему «слой» образуют знания, которые, хотя и могут быть переданы, но лишь в процессе длительной совместной работы: «Делай, как я!»

Далее по высоте «пирамиды», а значит, и по порядку уменьшения относительного объема лежит слой знаний, которые доступны для передачи в рамках традиционной педагогической процедуры: «могу попытаться объяснить, например, в рамках семестрового курса лекций и месячного лабораторного практикума, но не уверен, что все это можно формально описать».

И, наконец, едва различимая по относительному объему на фоне ниже лежащих слоев знаний, «верхушка пирамиды» — формализованные знания.

До сих пор объектом автоматизации на базе ЭВМ мог быть лишь самый верхний, исчезающе малый слой задействованных в реальном производственном процессе профессиональных знаний. В этом, видимо, и заключается основная причина относительно слабого влияния, которое успели пока оказать ЭВМ на макроэкономические показатели динамики хозяйственного механизма индустриально-развитых стран. Есть основания предполагать, что позитивные сдвиги в экономической эффективности внедрения ЭВМ в народное хо-

прикладной программы, в тех областях приложений, где задел формализованных знаний ранее полностью отсутствовал, становились, по мере массового распространения ПЭВМ, все более частым явлением, а в последнее время начали обретать контуры массового производственного процесса. Первый самый мощный слой профессиональных знаний оказывается при этом «уязвим» для проникновения в него «человечка с компьютером». Разумеется, «чужда» не происходит, и специалист, как и раньше, может «проникнуть» лишь в свою собственную «персональную зону» этого слоя, но ... выходит он из этой «зоны» нередко с работающей программой.

По общему характеру процесса разработки такая программа мало чем отличается от других «изделий», создаваемых в тех областях производственной деятельности, где уровень мастерства исполнителя слабо зависит от уровня владения им формальным аппаратом. По мнению итальянского ученого Дж. Атарди, процесс работы «непрограммирующего профессионала» за пультом ПЭВМ, характеризуется в первую очередь «приоритетом действия над планом». Как отмечал А. П. Ершов, это оказывается сегодня одним из наиболее характерных отличий «программирования для себя», которым заняты миллионы пользователей ПЭВМ от стиля работы значительной части профессиональных программистов, выполняющих, как правило, работу «на заказ». Регулярная произ-

Процесс записи ранее формализованных профессиональных знаний в готовой для непосредственного воздействия на машины и механизмы форме получил название — «программирование ЭВМ».

зайство, которые на макроэкономическом уровне станут заметны, как ожидают, к началу 90-х годов, будут связаны с феноменом «персональных вычислений» (personal computing).

Персональные вычисления

«Персональные вычисления» — это предоставленная миллионам людей возможность работать без посредников «один на один» с инструментом автоматизированной обработки информации. Отдельные «еретические эпизоды» в работе «непрограммирующихся профессионалов» за пультом персонального компьютера, завершаемые созданием реально полезной

водственная деятельность профессионального программиста по основной сути своей предполагает многоуровневое планирование и, соответственно, строго формализуемые условия «выполнения плана».

С другой стороны, объяснить (даже самому себе), как развивался за пультом ПЭВМ рабочий процесс, который через несколько десятков прошедших отладку и отвергнутых версий привел к единственному удовлетворяющему противоречивым производственным критериям варианту программы, «непрограммирующий профессионал», сознавая безнадежность ситуации, в большинстве случаев и не пытается. Принципиально новое качество такого «изделия» — программы ЭВМ перед любым дру-

гим сработанным аналогичным способом изделием заключается в том, что «динамика» функционирования системы или комплекса управляемых компьютером производственных агрегатов в этом случае может быть однозначно прочитана в «статике» исходного текста созданной программы. *Каким бы способом программа ни была разработана, если в конечном итоге опыт ее эксплуатации показывает, что она обеспечивает эффективный режим функционирования производственного оборудования, то это означает, что кроме непосредственного улучшения производственных результатов на данном рабочем месте получен и весьма важный «побочный» результат — формализованное описание найденного технического решения.* Понятно, что такого сорта «побочный продукт» может в ряде случаев оказаться сам по себе значительно более ценным, чем непосредственно наблюдаемый на данном рабочем месте производственный эффект внедрения ЭВМ.

Процесс автоформализации знаний и критерии «истинной науки»

В различных академических аудиториях нередко приходится сталкиваться с вопросом: «Зачем вам потребовалось для выделения некоторой совокупности элементов компьютерного творчества вводить новый термин — автоформализация? Это, ведь, кроме всего, еще и логический парадокс! Как может человек, неважно каким инструментом он пользуется, формально описать то, чего он не в состоянии объяснить даже самому себе? Наконец, согласитесь, что в ваших аргументах и доводах слышится много элементов, явно «потусторонних» по отношению к истинной науке.

Формулировка этого вопроса иногда меняется, но суть остается удивительно постоянной. Соответственно, и глава «Технология автоформализации профессиональных знаний» оказалась сразу же после выхода книги * пред-

лизиции профессиональных знаний, до сих пор воспринимались академически титулованными «жрецами-хранителями» его передовых форпостов как ересь, граничащая со святотатством.

Независимо от конкретного предмета и формы такого рода дискуссий, их концептуальная ось оставалась, как правило, дословно постоянной: «Математика ли это? Наука ли это?»

Мы, разумеется, не имеем здесь возможности сколько-нибудь подробно останавливаться на самых разнообразных иллюстрациях необычайно интересной для истории науки в целом устойчивости мифа о незыблемости «критериев строгой формализации» и их идентичности самому понятию научного метода. Изложение связанных с этим острых научных коллизий, блистательных взлетов и трагических тулупов, практических приложений можно найти в монографии советских математиков И. И. Блехмана, А. Д. Мышкина, Я. Г. Поновко «Механика и прикладная математика: Логика и особенности приложений математики». Анализ сложившейся ситуации с позиции истории развития математики и естествознания, обширный фактический материал для самостоятельных выводов содержит недавно переведенная на русский язык книга американского математика М. Клайна «Математика. Утрата определенности». Как отмечает автор во вступлении: «*Наши предшественники видели в математике непревзойденный образец строгих рассуждений, свод незыблемых истин в себе и истин о законах природы. Главная тема этой книги — рассказ о том, как человек пришел к осознанию ложности подобных представлений и современному пониманию природы и роли математики...*»

Кратко поясним общий смысл, вкладываемый в понятие автоформализации, простым примером. Пусть пульт бортовой ЭВМ установлен на гусеничном вездеходе и создан аппаратно-программный комплекс для автоматического управления таким «самодвижущимся» средством, включающий набор датчиков, воспринимающих окружающую обстановку.

Условия задачи: В центре труднопроходимого болота находится зимовье. Один из местных жителей иногда бывает там во время охоты и возвращается обычно без особых сложностей, так как умеет выбирать трассу сравнительно безопасного движения. По профессии он водитель и при необходимости мог бы провести к зимовью и вездеход.

Вопрос первый. Может ли бригада квалифицированных математиков и программистов, изучив предварительный алгоритм управления заданным транспортным средством, наблюдать затем в непосредственном общении за водителем во время его поездки на вездеходе к зимовью и соста-

...«применение математических методов в медицине, несмотря на относительно длинную историю, все еще находится в начальной стадии.

При первых же столкновениях с реальным медицинским материалом стало ясно, что те испытанные общие принципы, с которыми математики подходили к физическим и техническим задачам, в этой новой области плохо применимы. Аналогичное положение дел имеет место, по-видимому, и в других нетрадиционных для применения математики областях».

И. М. Гельфанд

Итак, кроме нового «компьютеризованного изделия», обеспечивающего более эффективный режим работы устройств и оборудования, *важнейшим результатом персональных вычислений оказывается зафиксированный на машинном носителе, готовый к тиражированию «формализованный фрагмент» из ранее принципиально недоступного формализации «нижнего слоя» индивидуальных знаний.* Вновь созданная «непрограммирующим профессионалом» программа может затем либо использоваться на одном или нескольких рядом расположенных рабочих местах, или, в зависимости от ее конкретной потребительской ценности, быть использована в качестве документа, специфицирующего условия правильности другой, функционально ей эквивалентной программы, заказанной для исполнения бригаде профессиональных программистов. Например, когда требуется повысить эффективность предназначенной для тиражирования программы по критериям машинных ресурсов, и т. д.

Процесс формализации профессиональных знаний, осуществляемый в режиме персональных вычислений, — исторически новая форма интеллектуальной деятельности. Поэтому нам представлялось целесообразным очертить круг связанных с этим творческим процессом объектов исследования специальным термином — автоформализация.

метод критики в значительной степени именно с этих «луристских» позиций. Коротко ответить здесь на все связанные с такого типа дискуссиями вопросы, разумеется, трудно. Поэтому попытаемся, не останавливаясь на эмоциональных оттенках критерия «научности», рассмотреть лишь отдельные, с нашей точки зрения, наиболее существенные аспекты обозначившегося здесь узла методологических противоречий.

Разумеется, ни ПЭВМ, ни какие-либо другие инструменты, не могут сами по себе резко изменить исторически сложившейся границы «зон определенности» профессиональных знаний. *Если мы говорим о том, что сегодня появляется возможность автоформализации профессиональных знаний средствами ПЭВМ, то должно быть ясно, что при этом существенно трансформируется и само понятие «формализации».* Судя по всему, это сегодня надо постоянно подчеркивать, чтобы избежать нередко возникающих в этом контексте терминологических недоразумений. Вместе с тем, любые попытки затронуть контуры «золотого храма» точных наук — монополии методов традиционной математики на процесс форма-

* Громов Г. Р. Национальные информационные ресурсы: проблемы промышленной эксплуатации. — М.: Наука, 1984. — 240 с.

вить в
автомат
по этом
Судя
мо, для
ющих в
для с
И дел
стях т
стка
Несмот
скачок
навигат
послед
профес
далеко
Более
отноду
лемами
Один
сорта
сионал
заказ
програ
Дж. Ф
обеспе
«Пр
нефте
«Экс
(Кана
фирма
долл.
моих
Как-то
сил и
ким об
включ
ления
«Очень
вет —
пробу
сарказ
Даже
тельные
седы с
нолог
лем, т
рирован
ний, м
грамм
новке
«квант

* Же
ла не
каци
ту или
задачу,
ком в
речь.
хьюсто
ники о
тающие
космич
в науч
назыв
многие
ваться
заводе
нии на
вестни
проект
посадк
ние ор
боратор

вить программу бортовой ЭВМ для автоматической проводки вездехода по этому же маршруту?

Судя по реакции зала, ответ, видимо, для большинства из присутствующих не оставляет заметных поводов для сомнения. Это невозможно! И дело, конечно же, не в особенностях того или иного конкретного участка труднопроходимой местности. Несмотря на резкий качественный скачок в оснащении, например, судов навигационным оборудованием, а в последнее время и бортовыми ЭВМ, профессия лощмана, увы, пока еще далеко не является анахронизмом. Более того, отмеченные трудности отнюдь не ограничены лишь «проблемами транспорта».

Один из ярких примеров такого сорта туфиков среди задач профессионального программирования «на заказ» приводит руководитель отдела программирования фирмы ИБМ Дж. Фокс в книге «Программное обеспечение и его разработка»:

«При попытке автоматизировать нефтеочистительные заводы фирмы «Эксон», расположенные в Эдмонте (Канада) и в Антверпене (Бельгия) фирма ИБМ потеряла более 10 млн. долл. Выполняли работу две сотни моих хьюстонских сотрудников. Как-то один из разработчиков спросил инженера компании «Эксон», каким образом он узнает, когда надо включать тот или иной клапан управления потоком в трубопроводе. «Очень просто,— услышал он в ответ.— Я опускаю палец в струю и пробую»... «Запрограммируйте это!»— саркастически заключает Фокс.*

Даже самые взаимно доброжелательные и творчески напряженные беседы с «лощманом», «инженером-технологом» или любым другим носителем, так называемых, слабоструктурированных профессиональных знаний, мало что могут прояснить программисту в содержательной постановке задачи. Дело, в том, что «квант времени» постижения существ-

ва сколько-нибудь нетривиальной из такого типа задач — жизнь... Поэтому на практике обычно возникает простая альтернатива: или повесить на трудноформализуемой прикладной задаче очередную стандартную «бирку» — «недозрела для автоматизации!» или поступить так, как рекомендовал поэт: «вот вам, товарищи, мое стило и можете писать сами». Попробуем проследить, как могли бы развиваться события в этом последнем варианте.

Предположим, что бригада занятых на «задаче о вездеходе» программистов вместо того, чтобы продолжать попытки алгоритмизовать неуловимые для непосвященных способы оценки «таежной ситуации», разработала необходимые базовые средства: драйверы для управления исполнительными устройствами навигационной системы вездехода, связи с датчиками окружающей обстановки и т.д., например, в рамках одного из популярных языков программирования высокого уровня и пригласила «охотника» за пультом бортовой ЭВМ, чтобы он сам попытался написать программу управления вездеходом, реализующую его собственный (неосознанный пока) «алгоритм проводки» транспортного средства к зимовью.

Предположим также, что «охотник» владеет основами «второй грамотности» и может начать работу за пультом ЭВМ, скажем, в рамках системы Бейсик. Сколько дней или недель для освоения конкретной версии языка с встроенными проблемно-ориентированными функциональными расширениями, и ... начинается процесс создания варианта программы для движения на начальном, простейшем участке трассы, а затем — долгий, изнурительный процесс отладки. Можно, видимо, в самых общих чертах представить себе, как это могло бы происходить.

«Совершенно естественно говорить об уме более интуитивном, когда зона комбинирования идей находится глубоко, и об уме логическом, если эта зона расположена достаточно поверхностно».

Ж. Адамар

«Пуск!» — машина продвинулась на несколько метров и провалилась в трясицу. Подъем, буксировка вездехода на исходную точку трассы, анализ ситуации. Автор винмательно осматривает местность вокруг гусеничного следа, вплоть до того участка, где эти следы исчезают под водой, долго водит пальцами по листингу программы. «Так ... кажется, ясно. В программу была заложена неполная информация: выбор направления движения с ориентировкой на более сухой мох осуществляется лишь в пределах зоны «талой воды». В случае, когда недавно прошел дождь, следует дополнительно учитывать также и цвет развода и ориентироваться в движении на более «рыжие» участки, где обычно грунт оказыва-

ется плотнее». В текст программы вносятся необходимые изменения, предварительная отладка на машинном макете — «тексткарте» местности, и снова прогрев двигателя, команда «Вперед!» На этот раз машина прошла чуть дальше, и т.д.

Необходимый инструментарий, оценки эффективности, границы применимости...

Как и всякий чисто умозрительный, иллюстративный пример* рассматриваемый случай одинаково уязвим со всех сторон.

«Представить себе, чтобы «человек из тайги» мог на время отложить ружье и позабавиться с пакетом игровых программ — это еще куда ни шло... но, чтобы он написал программу реального времени...?» — резонно усомнится один.

«А не потребуются ли и для этого, как Вы его называете, процесса автоформализации, все тот же «квант времени», длиной в жизнь?» — сочувственно улыбнется другой.

И, наконец, третий решительно «закрывает проблему» роковым вопросом: «А Вы уверены, что управляемая такой программой машина после скажем, двух-трех удачных рейсов пройдет трассу еще хотя бы раз? В другое время суток? При другой погоде?..»

Первый вопрос, видимо, адресован в значительной степени профессиональным программистам, которые создают базовые программные средства для инструментального обеспечения процесса автоформализации знаний. В этом контексте уместно еще раз отметить важность дальнейшего развития концепции «объектно-ориентированного программирования», согласно которой еще до первого контакта пользователя с ПЭВМ должен быть создан проблемно-ори-

ентированный базовый инструментарий, который обеспечивает конечному пользователю регулярную возможность самостоятельно программировать достаточно сложные производственные процессы, в том числе и процессы реального времени, совершая лишь самые необходимые и ес-

* Конкретный пример реализации одного из вариантов технологии автоформализации профессиональных знаний в рамках практически решаемой задачи автоматизации биотехнологических исследований рассматривается в упомянутой выше книге: «Национальные информационные ресурсы: проблемы промышленной эксплуатации», с. 139—143.

* Желая подчеркнуть, что суть дела не в том, достаточно ли квалификации у программиста, чтобы решать ту или иную трудноформализуемую задачу, Фокс особо оговаривает, о ком именно в данном примере идет речь. Упоминаемые им «две сотни хьюстонских сотрудников» — сотрудники отделения фирмы ИБМ, работающие по контракту с управлением космических исследований (НАСА) в научном центре Хьюстона. Это, так называемые, «суперпрограммеры», многие из которых до того, как заняться «задачей о нефтеперегонном заводе», принимали участие в создании наиболее сложных из всех известных за рубежом программистских проектов: программное обеспечение посадки человека на Луну, управление орбитальными космическими лабораториями и др.

тественные, с точки зрения ранее накопленного профессионального опыта, действия «в мире информационных объектов», отображаемых на экране ЭВМ. Процесс такого «интуитивно-го» программирования приближается к процессу управления развитием сюжета в компьютерных играх*.

Суть второго вопроса можно было бы, видимо, дополнительно пояснить известным утверждением, что, в принципе, каждый человек мог бы стать академиком, только одному для этого потребуются тридцать лет, а другому — триста... Разумеется не всякую прикладную задачу и далеко не всякий «непрограммируемый профессионал» сумеет формализовать средствами персональных вычислений. Как заметил один из японских экспертов, «продавать ПЭВМ, говоря при этом, что можно получить с помощью ПЭВМ, это то же самое, что продавать авторучки, утверждая, что с их помощью можно написать нечто на уровне премии Акутагавы или Нобелевской премии».

Специалист в данной предметной области, вооруженный средствами ПЭВМ для формальной регистрации совершаемых им действий, может при определенных условиях зафиксировать в кодах машины какой-то след выполняемого им производственного процесса. Нередко это оказывается возможным и в тех случаях, когда проникнуть каким-либо иным способом извне в его «творческую кухню» оказывается практически невозможным. Однако, нет и не может быть какой-либо «предопределенности свыше» успеха или провала каждой отдельно взятой попытки автоформализации профессиональных знаний. Все, чем мы здесь располагаем сегодня, — это опыт, накопленный миллионами пользователей ЭВМ за последние десятилетия компьютерной эры, который со всей определенностью показывает, что вероятность успеха в решении трудноформализуемых производственных задач оказывается, как правило, значительно выше при движении профессионала в данной предметной области от объекта автоматизации к ЭВМ, чем наоборот — профессионального программиста от ЭВМ к автоматизируемому процессу. Еще в начале 70-х годов, с самых первых попыток организовать промышленное внедрение средств автоматизации на базе микроЭВМ, американские эксперты в области микропроцессорной техники вынуждены были признать: «для нас труднее понять, как работает автомобильная фирма, чем для них — как работают наши микропроцессоры».

И, наконец, третий вопрос. Разумеется, нет никаких гарантий в том,

что созданная «охотником» навигационная программа обеспечит в какой-либо очередной раз благополучную проводку вездехода, сколько бы не состоялось предварительно успешных рейсов! Более того, именно поэтому мы и выбрали для иллюстрации столь экзотический пример, чтобы более ясно выделить границы рационального использования методов автоформализации знаний.

Дело в данном случае даже не в том, что любые сколько-нибудь сложные программы в новых областях

алгоритм своего поиска в машину, причем так, что созданная им программа в 60 % случаев позволяет немедленно получить нужный документ, значит на данном рабочем месте оказалось возможным заметно повысить производительность труда. Понятно, что в остальных 40 % запросов, когда программа «отказывает», никакой «аварии» не произойдет, а будет выполнен «псевдоручной» поиск. В то же время даже малая вероятность «отказа» программы проводки упомянутого выше вездехода непрям-

Вся область профессиональной человеческой деятельности, которая принципиально поддается пока формализации, а, значит, и автоматизации на базе ЭВМ — это, образно говоря, тонкая поверхностная пленка формализованных знаний, лишь слегка прикрывающая поверхность океана накопленного человечеством неформального знания.

приложений, как правило, начинают «спотыкаться»... Суть вопроса в данном случае в другом. Ведь и в метро со всей его максимально достижимой стабильностью трассы и постоянством метеословий никто, как известно, не решился еще снять споезда машиниста... Автоматика ему только помогает. Понятно, поэтому, что областью приложений методов автоформализации профессиональных знаний должна быть лишь так называемая безкризисная зона производственной деятельности человека. Что это такое? Поясним на простом примере.

Пусть создана база данных для машинного хранения и оперативного поиска документов, которые ранее хранились в рабочих столах исполнителей. Предположим, что обращение пользователя ПЭВМ за необходимым документом происходит с помощью популярной сегодня «метафоры рабочего стола», т. е. пользователь видит на экране ЭВМ условное изображение разного типа «ящиков», из которых он может, управляя курсором, «извлекать» изображения «канцелярских папок» и т. д. вплоть до нужного ему документа. Методами «объектно-ориентированного программирования» пользователь может запрограммировать различные алгоритмы поиска нужной ему информации, в том числе и весьма близко приближающиеся к тем «эвристикам», которые он использовал в «домашнюю эру».

Многим знакома просьба сотрудника, который должен на какое-то время оставить свое рабочее место: «Пожалуйста, не трогайте моих бумаг, а то я потом ничего не найду!» Как же находит обычно человек в хаосе доверху забитых самыми разными бумагами ящиков своего письменного стола нужный ему документ? Мало кто, видимо, пытался всерьез задавать себе такие вопросы. Но, если тот же сотрудник, вооруженный ПЭВМ, смог по итогам длительного процесса автоформализации заложить

лема из-за «кризисной ситуации», которой чреват такой отказ.

Если доля ручных операций на каком-либо рабочем месте в результате автоформализации профессиональных знаний исполнителя снизилась, скажем на 30...40 %, то это в большинстве случаев уже вполне заметный прирост производительности труда, который, как правило, полностью окупает затраты на установку ПЭВМ. Тот факт, что в остальных 60...70 % случаев программа «пасует» и оператор вынужден, как и раньше, брать управление на себя, означает лишь, что пока не удалось на данном рабочем месте обеспечить учет большего числа факторов сложного производственного процесса. Традиционная альтернатива «все или ничего» в оценке работоспособности прикладной программы в данном случае не уместна, так как отношение числа успешных срабатываний к «отказам» оказывается лишь текущим показателем достигнутого выигрыша в росте производительности труда. Принципиально новая ситуация складывается, когда «отказ» связан с возможностью аварийного исхода. «Кризисная область» приложений ЭВМ и должна быть поэтому основным объектом так называемого «доказательного» программирования.

Итак, для четкого размежевания областей приложений математически безупречного «доказательного» программирования, выполняемого исключительно профессиональными программистами, и процесса автоформализации знаний, в который постоянно вовлекается все большее число «парaprogramмистов» — самого массового контингента пользователей ПЭВМ, видимо, достаточно вспомнить три известных закона робототехники, сформулированные А. Азимовым: 1) робот ни при каких обстоятельствах не должен нанести вред человеку; 2) робот не должен наносить вред себе или другим роботам в тех случаях, когда это не противоречит первому закону робототехники;

3) робот должен противостоять человеку. По программе формальной, «третьей» Обескуражен «первый» снизит это до максимума строго на за...

Верно! Пусть темой программ транслируется условия движения больше простые малые до ввематизации карьеры и...

Иные шаемс. Только провер «закон» бот обесне средств рые дность на как столкн как те движел выполне объект нзя сн электр чтобы кой ма комьюк

Пусть програ маршрут значит, 70 % ности нечить рые о бригад дут те находя старта Когда но усд того, ч ного м 60 %, з быть в водител

Посл ся, что работав вычисл

* Подробнее с этим подходом можно познакомиться, например, в статьях А. П. Еришова и Г. В. Лебедева в последних номерах «МП».

3) робот выполняет любое указание человека в тех случаях, когда это не противоречит первым двум законам.

Понятно, что эксплуатация программ, полученных методами автоформализации профессиональных знаний, допустима лишь в рамках «третьего закона робототехники». Обеспечить безусловное выполнение «первого закона» и, по возможности, снизить риск нарушения «второго» — это профессионально выполняемая с максимально достижимым уровнем строгости задача «программирования на заказ».

Вернемся к «транспортной задаче». Пусть в цехе с многоуровневой системой автоматизации производственных процессов, необходимо запрограммировать траекторию движения транспортного робота так, чтобы в условиях частой смены маршрутов движения и одновременной работы большого числа таких роботов время простоя станков в ожидании доставляемых заготовок было бы минимальным. Предположим также, что до введения системы «сплошной автоматизации» в цехе работали электрокары с опытными водителями, которые исключали простой оборудования.

Иными словами, мы снова возвращаемся к задаче об «охотнике»... Только начать теперь надо будет с проверки выполнения первых двух «законов Азимова». Транспортный робот должен быть предварительно обеспечен аппаратно-программными средствами «нижнего уровня», которые полностью исключают возможность совершить наезд на человека, на какое-либо цеховое оборудование, столкновение роботов и т. д. В рамках тех возможностей управления движением, которые остаются после выполнения «условий безопасности», объектно-ориентированная программная система предоставляет водителю электрокара необходимые средства, чтобы экспериментировать с прокладкой маршрута, например, в режиме компьютерной игры.

Пусть в первом варианте такой программы робот успешно проходит маршрут лишь в 40 % случаев. Это значит, что в первом приближении 70 % ранее существовавшей численности водителей смогут теперь обеспечить тот же поток деталей, которые они раньше доставляли всей бригадой (предполагая, что 10 % будут теперь заняты лишь тем, чтобы находить и возвращать в точку старта «заблудившихся роботов»). Когда программа будет дополнительно усовершенствована и вероятность того, что робот не сойдет оптимального маршрута, достигнет, скажем, 60 %, это будет означать, что может быть высвобождена почти половина водителей и т. д.

Последний пример нам потребовался, чтобы еще раз подчеркнуть: разработанная в режиме персональных вычислений программа должна оце-

ниваться по совершенно иным критериям, чем профессионально создаваемый на заказ «программный продукт», потребительская ценность которого вообще не может обсуждаться до получения «сертификата» о безошибочном прохождении комплекса узаконенных тестов. С другой стороны — продукт автоформализации профессиональных знаний часто оказывается *локально полезным* еще задолго до окончательной доводки со стороны ответственной программы. Более того, для многих из такого типа программ сама по себе постановка задачи на достижение близкого к 100 % уровня надежности может оказаться экономически просто бессмысленной, в то время как программа, которая надежно «срабатывает» даже менее, чем в половине практически интересных случаев, может дать весьма ощутимый прирост производительности труда на автоматизируемом рабочем месте.

Разные задачи, разные технологии, разные области приложений и, как следствие, принципиально различные критерии качества процесса формализации профессиональных знаний.

Понятие формализации наполняется, таким образом, новым «инструментальным» смыслом и далеко выходит за те жесткие дедуктивно-логические рамки, в которых оно сформировалось за первые 25 веков развития доминирующей сегодня «антической ветви» науки математики.

Успех или неудача акта формализации знаний все более нередко определяется не уровнем его логической доказательности, а принципиально иными «прагматическими» критериями (например, как показано выше, это могут быть... экономические критерии). Иными словами, с расширением областей приложений ЭВМ типичными становятся ситуации, когда традиционно неразрывная со временем «программы математизации знаний Пифагора из Самоса» концептуальная связь двух фундаментальных понятий: формализация и логическая доказательность — не сохраняется. Именно это обстоятельство и имело в виду, когда мы отмечали выше, что с массовым переходом к технологии автоформализации знаний трансформируется само понятие — формализация.

Инверсная «триада»

Профессионал в данной предметной области, как правило, знает, как найти верное решение практически в любой конкретной ситуации, которая может возникнуть в ходе реального производственного процесса, однако не знает и, видимо, не может знать, какие именно рабочие ситуации и в какой последовательности сложатся на очередной «трассе решений».

Главное внешнее отличие режима «персональных вычислений» от традиционных методов программирова-

ния заключается поэтому в том, что к моменту пуска первого варианта программы ее автор не может быть уверен в том, что знает верный путь решения. Все, чем он располагает, — основанная на профессиональном опыте уверенность, что выход из любой конкретной ситуации будет найден непосредственно в контексте решения («по месту»). В значительной степени с этим обстоятельством и связана принципиальная сложность формализации интуитивно, видимо, существующих решений «слабоструктурированных» задач. Непреодолимая, как правило, трудность на пути формального описания такого типа задач — необходимость учета слишком большого числа факторов. Относительный вес этих факторов меняется от реализации к реализации заранее непредсказуемым образом... Именно это последнее обстоятельство ограничивало до сих пор возможность использования в решении на ЭВМ такого типа прикладных задач стандартной последовательности этапов: математическая модель, алгоритм, программа (типичная «триада» решения).

Моделирование предлагает, как известно, возможность *предварительно* выделения из большого числа факторов, реально влияющих на решение любой практической интересной задачи, относительно небольшого числа наиболее важных. Это и позволяет абстрагироваться при изучении математической модели от сложности всего остального «окружения» реального мира. Понятно, что в задачах, где априори невозможно выполнить разделение большого числа переменных на «существенные» и «несущественные», не может быть использован этот традиционный механизм абстракции. Приходится либо искать какие-то другие подходы к решению прикладной задачи, либо ... объявить такие задачи «донаучными» (выбор удобных здесь эпитетов необычайно богат), т. е. закрыть проблему старым, как мир, и поэтому безусловно надежным приемом — «зелен виноград»...

Альтернативный подход — автоформализация профессиональных знаний — позволяет, при необходимости, вернуться к математическому исследованию автоматизированного средствами персональных вычислений производственного процесса. В ряде случаев оказывается возможным использовать машинную выдачу текста созданной «непрограммирующим профессионалом» программы для реконструкции по ней хотя бы самых общих контуров неосознанно используемого автором алгоритма управления. При этом может быть сделана попытка использовать «восстановленный» алгоритм как базу для создания методами технологии программирования «на заказ» более эффективной по машинным ресурсам програм-

Основные этапы развития технологии знаний (оценка автора)

Знания				Производство			
Основные вехи информационной технологии	Время от наших дней (годы)	Общая характеристика этапа технологии	Уровень технологии знаний	Доминирующий тип производственной культуры	Типовой производственный процесс	Темп производственного роста	
						По валу: рост общего числа единиц выпускаемой продукции	По номенклатуре: рост разнообразия типов выпускаемой продукции
Наскальные изображения Письменность	30—20 тыс. 6 тыс.	Домашний этап: первые инструменты счета из кости и камня, календарь, доменные, водяные и т. д.) часы, компас, бумага, книги	Ручная технология регистрации знаний	Созерцательная: статически-художественная	Ремесленное производство уникальных и мелкосерийных изделий	Медленный	Медленный
Книгопечатание	500	Первая информационная революция					
Телеграф Фото Радио ЭВМ	150 » 90 40	Машинный этап: печатный станок, механические часы, телеграф, фото, телефон, фонограф, радио, кино, магнитозапись, телевидение, ЭВМ	Машинная технология тиражирования и распространения знаний	Механистическая: формально-логическая	Механизированное массовое производство стандартизируемых изделий	Быстрый	Медленный
Персональные ЭВМ	10	Вторая информационная революция					
?	?	Симбиотический этап: «игровая компонента» ПЭВМ, индивидуальные рабочие станции, локальные сети, системы индивидуального доступа к отраслевым, региональным и мировым информационным ресурсам	Человеко-машинная технология автоматизации знаний	Игровая: динамически-художественная	Гибкое автоматизированное производство уникальных и мелкосерийных изделий	Быстрый	Быстрый*
?	?	Третья информационная революция					

* Примечание: Впервые оказывается, что машина в состоянии помочь человеку в расширении пределов его фантазии, поиске новых богатств разнообразия, а не только в увеличении физических возможностей к многократному повторению чего-то однажды найденного. «Детонатором» этого процесса стала так называемая игровая компонента ПЭВМ. Подробнее об этом см. «Игровая компонента — первое функциональное отличие персональных ЭВМ», — журнал «Микропроцессорные средства и системы», 1984, № 1, с. 46—47 или этот же раздел в упомянутой выше книге «Национальные информационные ресурсы...», с. 124—127.

мы, исходность когда лесобор жиров «персона гическо Нако задача дельны наприи участка может алгорит уровни углубле создаи шить « абстрав дель — напраи И так вая «1 цептуа: от того мости дачи, т «восход где это граммн как тр ху-вниэ случае неприа триадаз ведут усложн интегр вверха решени ет пред ной « Прои зации однона было в чально «АСУ-и эта воэ чески в странах стойкое заказч там» в нологии все-таки что изб сложней таюцей цеха и него из функций мальны зации, «враста в «жисе Общи

* Нас ствия» вески которм и внутр ной час

мы, функционально эквивалентной исходному «макету». Такая потребность может возникнуть, скажем, когда оказывается экономически целесообразным начать массовое тиражирование найденного в режиме «персональных вычислений» технологического решения.

Наконец, в случаях, когда ставится задача объединить между собой отдельные «островки автоматизации», например, отдельные технологические участки, в автоматизированный цех, может потребоваться для разработки алгоритма управления следующего уровня поставить задачу еще более углубленного исследования ранее созданной программы — т.е. завершить «движение» по трем уровням абстракции исходной задачи: модель — алгоритм — программа — в направлении «снизу-вверх».

Итак, с нашей точки зрения, типовая «триада» не утрачивает концептуальной целостности независимо от того, используем ли мы, в зависимости от характера решаемой задачи, технологию «нисходящего» или «восходящего» проектирования. Там, где это возможно, разработчик программного комплекса идет в рамках традиционной технологии «сверху-вниз» от модели к программе. В случае когда этот путь оказывается неприемлем, используется «инверсная триада» и проектирование комплекса ведут путем его последовательного усложнения от частных решений к интегрированной системе «снизу-вверх». Понятно, что конкретный путь решения целиком и полностью диктует предметная область, а не тот или иной «догмат академической веры».

Проектирование систем автоматизации производства исключительно однонаправленно — «сверху-вниз» — было наиболее характерно для печально памятного процесса всеобщей «АСУ-низации». К началу 80-х годов эта волна прошла*, оставив практически во всех промышленно развитых странах многомиллионные убытки и стойкое отвращение разработчиков и заказчиков ко всякого рода «догматам» в компьютерной науке и технологии. Весьма дорогой ценой, но все-таки удалось при этом выяснить, что избежать «реакции отторжения» сложившегося организма давно работающей организации (учреждения, цеха и т.д.) в ответ на внедрение в него извне «чужеродного тела» — функционирующей по строго формальным законам системы автоматизации, можно лишь при поэтапном «растании» ее отдельных подсистем в «живое тело» организации.

Общий смысл нового подхода: «до-

* Наследием этого «стихийного бедствия» остаются многочисленные вывески разного типа «АСУ-контроль», которыми все еще оклеены фронтоны и внутренние двory зданий центральной части многих наших городов.

рожки сначала протоптаются, а потом уже асфальтируют...» В начале 80-х годов вместе с ПЭВМ появилась и концепция «архипелага автоматизации». Смысл ее американские эксперты сформулировали просто: use now, integrated later (используйте ЭВМ сразу, объединяйтесь в сети, многоуровневые структуры — потом...). Сначала ЭВМ должна стать неотъемлемой, органичной частью действующей структуры организации, должны возникнуть и доказать свою локальную полезность «островки автоматизации» на базе компьютерной технологии и только затем уже можно будет поэтапно ставить вопрос о наиболее рациональных способах их объединения в иерархию формально взаимодействующих систем.

Секрет особой «живучести» эволюционно усложняющихся систем, по сравнению с циркулярно насаждаемыми АСУ, заключается, кроме прочего, и в том, что в их структуру и алгоритм функционирования оказываются «запаяны» методами автоформализации многолетний опыт и профессиональные знания сотрудников о наиболее рациональных (а в некоторых случаях и уникальных) для данной организации способах формирования информационных потоков и управляющих воздействий.

В большинстве практически интересных случаев предварительное изучение предметной области позволяет оценить достижимый уровень определенности для совокупности внешних факторов, влияющих на условия конкретной задачи. Это создает необходимые предпосылки, чтобы на одном из первых этапов проектирования выбрать соответствующий уровень требований к внутренней логической строгости решения. Тем не менее, до самого последнего времени принято было считать, что независимо от степени исходной определенности условий, процесс решения задачи должен отвечать традиционным со времен «античной математики» требованиям логической точности выводов.

Давно уже считается общепринятым, что уровень точности числовых выкладок должен соответствовать реально достижимой в конкретной задаче точности регистрации исходных данных. В то же время попытки распространить тот же, по-существу, принцип «равнопрочности» основных этапов процесса преобразования исходной информации на уровень логической строгости ее решения воспринимались, в лучшем случае, как нарушение неписаных правил хорошего математического тона, а чаще, как проявление логической небрежности, поощрение математической «полудрамотности» и т.д. Было бы уместным, видимо, еще раз напомнить здесь известное предупреждение Ф.Энгельса о том, что «если захочешь добить-

ся математической достоверности в вещах, не допускающих этого, нельзя не впасть в нелепость или в варварство».

Если мы ясно сознаем, что не в состоянии учесть на требуемом уровне логической строгости внешние факторы, влияющие на условия задачи, то в какой степени должна нас заботить логическая безупречность внутренней схемы ее решения? Рациональным было бы, видимо, полагать, что вся трасса решения задачи от условия до результата должна быть «логически равнопрочной». Иными словами, чем больше внешней неопределенности в условиях, тем менее обоснованными оказываются строгие требования к внутренней логической доказательности формализуемого решения. И наоборот, для класса задач с гарантируемой степенью логической определенности условий требования «доказательного», логически безупречного вывода алгоритма ее машинного решения оказываются рационально обоснованными, естественно вытекающими из существа «поставки задачи».

Интуиция и формальная логика: дуализм творческого процесса

Следует особо отметить, что задача изучения характера взаимодействия интуитивной и логической компонент в творческом процессе (главным образом, в рамках «таинства» научного творчества) привлекала внимание исследователей еще задолго до того, как с появлением «феномена ПЭВМ» эта проблема оказалась в ряду наиболее актуальных, определяющих методологические основы массовой компьютеризации.

В 1969 г. советский психолог М. Г. Ярошевский высказал предположение о функциональном дуализме этих двух традиционно противопоставляемых компонент. По его мнению, «имело бы смысл применить нечто сходное с «принципом дополтельности» в физике к трактовке отношений между формализуемыми (объективно отчуждаемыми от субъекта) и неформализуемыми, интимноличностными, неотчуждаемыми от субъекта, компонентами творчества. Не утратило ли бы тогда свою антигеничность противопоставление интуитивного акта, относимого к сфере психологии, формализованной операции, относимой к логике, пронизывающее на протяжении веков учение о творчестве?»

В качестве примера, иллюстрирующего характер взаимодействия этих компонент творческого процесса при решении «вершинных» задач человеческого интеллекта, М. Г. Ярошевский ссылается на фрагмент беседы Эйнштейна с одним из психологов, занятых «картографированием» трассы великих научных открытий: «В тече-

ние всех этих лет,— вспоминает Эйнштейн.— Имелось чувство направления, движения вперед по направлению к чему-то конкретному. Очень трудно, конечно, выразить это чувство в словах, но оно безусловно имело, и оно должно быть отделено от последующих размышлений о рациональной форме решения. Конечно, позади этого направления всегда имело нечто логическое, но я имел его в виде обзора в зрительной форме».

Есть, видимо, основания предполагать, что для всех обозримых сегодня «эшелонов» научных исследований, профессионального искусства или интеллектуального ремесла (к последним относят, например, и повседневную практику программирования): от исторически признанных лидеров того или иного этапа развития науки до рядовых специалистов выполняется общая закономерность развития творческого процесса. В задачах относительно простых для данной предметной области формальная логика «прокладывает» трассу решения, для наиболее сложных — обочивает пройденный путь.

Ведущая тенденция, которая сложилась уже в первые десятилетия компьютерной эры в «разделении труда» между двумя «партнерами» по диалогу «человек—ЭВМ», просматривалась достаточно четко: человек решает наиболее «интуитивно нагруженную» часть производственной задачи, а машина — формально логическую. Однако трудность в практической организации режима массового использования такого «диалога» заключалась в том, что вся тяжесть задачи предварительного «вычленения» формальной компоненты из всей совокупности профессиональных знаний в данной предметной области ложилась обычно на программиста. Принципиально непреодолимый характер возникающих при этом трудностей в тех наиболее интересных случаях, когда прикладная задача не имела ранее созданного «математического каркаса» решения, выше мы уже обсуждали.

«Феномен ПЭВМ» позволил сделать шаг к преодолению этого противоречия. Оказалось, что может быть создан инструментарий, который позволяет не только исполнять на ЭВМ формально поставленные задачи, но и помогать человеку в свершении «интимного» акта вычленения формализуемой компоненты из его индивидуальных профессиональных знаний. Необходимые для повышения эффективности такого процесса научные методы и технические средства образуют в совокупности «технологии автоформализации профессиональных знаний».

Еще недавно доступные только «богоязычникам» единицам (да и то далеко не в любой профессиональной области) возможности не только решить прикладную задачу, но и фор-

мализовать затем найденную схему решения, становятся сегодня в режиме «персональных вычислений» рутинной для миллионов пользователей ПЭВМ рабочей процедурой. Вряд ли требуют особых комментариев нередкие высказываемые в этой связи на страницах массовой печати рассуждения отдельных футурологов из процветающей ныне когорты технократических «экстремистов», которые всерьез обсуждают влияние «века информатизации» на процесс вызревания новых открытий или темпы рождения современных «платонов, ньютонов, ломоносовых и эйнштейнов». Однако было бы в некотором смысле «симметричным» заблуждением не замечать наблюдаемого сегодня ускорения темпов «дренажа» тонких методов творчества, традиционно связываемых с понятием «интеллектуальной элиты», в область массового «рабочего творчества». «Феномен ПЭВМ», с которым в первую очередь связывают сегодня «демократизацию» научного творчества, — только одна из наиболее заметных вех этого процесса.

ЗАКЛЮЧЕНИЕ

Точка зрения автора на общий характер исторического процесса развития технологии знаний представлена таблицей (см. с. 88).

Главные отличительные черты «второй информационной революции» в наиболее краткой их формулировке, видимо, могут быть сведены к трем основным пунктам:

1. Взрывной процесс «демократизации» научного творчества — все более широкое вовлечение в процесс формализации профессиональных знаний миллионов трудящихся самых различных специальностей, образовательного уровня и индивидуальных интересов.

2. Резкое ускорение технологического цикла развития* ведущих отраслей общественного производства — активное включение формализованных профессиональных знаний непосредственно в производственный процесс (например, когда эти знания обретают форму прикладных программ, управляющих станками с ЧПУ, технологическими участками, системами автоматизации научных экспериментов, производственных испытаний, обработки текстов и т. д.), минуя необходимую при «книжном тиражировании» стадию опосредованного воздействия на человека-исполнителя.

3. Массовое тиражирование накапливаемых профессиональных знаний в масштабах, сопоставимых с теми, которые ранее обеспечивал только печатный станок; однако, в отличие от эры книгопечатания, хранимых в

* Имеется в виду упомянутый выше регенеративный цикл: знание — производство — знание (см. с. 80).

готовой для автоматизированного поиска «машинной форме», что создает необходимые технические предпосылки для постоянного расширения сферы практического использования индустриальных методов человеко-машинного производства новых элементов профессионального знания.

Отметим в заключение, что ПЭВМ, как и любой инструмент, лишь создает определенные технические предпосылки для повышения эффективности творческого процесса, но не в состоянии его вызвать.

Это уже наша с вами задача — воспитать поколение людей, для которых работа за пультом ПЭВМ в режиме автоформализации профессиональных знаний будет одной из наиболее естественных форм творческого самовыражения и, создать необходимые социально-экономические условия для постоянного стимулирования потребности в таком самовыражении (понятно, что именно последнее условие является и наиболее важным и наиболее трудновыполнимым).

В 1984 г. Д. Кнут, выступая с «Тьюринговской лекцией», предложил свой вариант определения границы между наукой и профессиональным искусством:

«Наука — это та часть наших знаний, которую мы сумели понять настолько хорошо, что можем обучить этому ЭВМ. Там, где мы еще не достигли такого уровня понимания, речь пока идет лишь о профессиональном искусстве. Формальная запись алгоритма или программы ЭВМ, по существу, позволяет нам выполнить весьма полезный тест глубины наших знаний, так как переход от искусства к науке просто означает, что мы поняли, наконец, как автоматизировать данную предметную область».

Методы автоформализации оказываются основным средством включения в активный «производственный» фонд научного знания новых и все более мощных пластов профессионального искусства. Научная разработка и широкое практическое использование этих методов — важнейший фактор ускорения процесса массовой компьютеризации народного хозяйства.

КРАТКО ОБ ОТВЕТАХ НА ВОПРОСЫ

Все поступившие за время доклада записки с вопросами можно было бы, видимо, условно разделить на три заметно неравные группы.

Первая и самая большая группа — это записки с конкретными вопросами к тому или иному разделу доклада. Ответы на них, как правило, непосредственно встраивались в «текст изложения» по мере их поступления, что, возможно, приводило к некото-

рым исключениям, что в чужим правилам не принято.

Вторые — это вопросы, касающиеся критериев уровня области «матрикс» ранжирования, в которой надо жить, чтобы быть актуальными. И ясно, что на основе семинара защиты докторской диссертации в аудиторе

В настоящее время нам не замечено

«Вот, дело для профессора в том, дела для профессора? И, и понять, сумеет ли просто, здоровым

И, на наш взгляд» и Остановившись на значительных традициях знаний

С нашей наобороте, конечно же «полностью» ставленными следует расширить. Значительные методы профессионального пытались класть на головы исполнителей в высшей степени. Наконечник текста е централизованной автоформализации не

Хочу обратить Ваше внимание на опасность массового распространения «компьютерной полуграмотности». Я имею в виду не недостаточное качество обучения на сегодня, я знаю Вашу точку зрения, что важно хотя бы начать. Вопрос в том, что многие готовы были бы считать вопрос о грамотности решенным (и «закрытым») в результате обучения чему-нибудь вроде Бейсика. И вопрос не в том, что Бейсик плох (или, может быть, хорош), а в том, что это совсем не то умение, которое поможет массовому специалисту, нуждающемуся в использовании вычислительных средств. Не потому, что не нужно в школе знать какой-то язык типа «мини-языка» Информатики-9 (не знаю, как его называть, но свято место пусто не бывает, и злые языки уже прозвали его «Ершолом»). Только этот язык (как и формируемое на его основе представление об алгоритме), скорее, относится к математическому курсу, а не к информатике (аналогично знанию прогрессий или символа Σ).

Всерьез специалисту нужно не кропать для себя мелкие самостоятельные программки (их возможности, особенно у непрофессионала, весьма ограничены), а уметь обращаться к готовым средствам, дополняя их по своему собственным программированием. Но против чего я больше всего здесь возражаю, это против ориентации на «инстинктивное» программирование, по Г. Р. Грому. Думаю, что в сложном случае человек при таком подходе просто не справится с задачей и махнет рукой на программирование вообще.

Появилась и другая серьезная опасность. Старшие школьники и студенты, «дорвавшиеся» до машины и предоставленные самим себе, замыкаются в примитивном круге задач, превращаются в «фанатиков» (термин моих студентов, наблюдавших это явление в свою беготню в школе-интернате). Их больше всего начинает интересовать хитроумное использование каких-либо малоизвестных возможностей машины или операционной

системы, возможность обойти разные запреты, недобро подшутить над соседом. Я уж не говорю о пролезании на машину с черного хода или ночных бдениях. Замкнувшись во всем этом, иные утрачивают способность перейти к более серьезным задачам, вылетают с факультета, а то и попадают в больницу. Или такой «пострел» уже на первом курсе начинает наводить справки, у кого бы купить за наличные бобину магнитной ленты (краденую, естественно). О заграничной породе «хакеров» (hackers) Вы, вероятно, лучше меня знаете. Это явление, может быть, не специфично для программирования: В. А. Залгаллер, учивший нас в университете геометрии (и заодно уму-разуму), рассказывал об опасности для способного школьника или студента застрять в круге задач элементарной математики.

И не стоит отвечать, что все эти ребята со временем «перебесятся», зато приобретут необходимую в других делах виртуозность. Я никак не против игры и не против виртуозности, но если мы не поведем этих людей вперед, ставя перед ними такие задачи, по сравнению с которыми всякие машинные трючки покажутся им неинтересными, то само это не пройдет. И не надо ссылаться на наше поколение, успешно миновавшее эту фанатизацию. Вот аналогия: моего сына кое-кто упрекает в злоупотреблении сладкой пищей. Я попытался вспомнить, почему мне такое не повредило. И вспомнил: денег не хватало.

Здесь о молодежи я сказал то же, что о взрослых, ликвидирующих «вторую безграмотность». Нельзя останавливаться на элементарных навыках программирования, а может быть, и вообще не стоит делать на них упор.

Доктор физ.-мат. наук Г. С. Цейтин,
математико-механический факультет
Ленинградского государственного университета

Григорий Самуилович!

Разделяя в целом Ваши предостережения об опасности «компьютерной полуграмотности», хотел бы сделать ряд замечаний в защиту различных сторон компьютерного всеобуча.

Мы воспринимаем термин «массовая культура» всегда с некоторыми оговорками, не упуская ни одного повода поговорить об ее издержках. Однако уже в силу объективных законов социальной психологии мы не можем от любого крупномасштабного явления оставить только его одну, «лучшую» сторону в надежде избавиться от другой, «худшей».

Полуграмотность, бескультурье, злоупотребление, сектантский фанатизм — это, к сожалению, универсальные стороны человеческого бытия, для которых предметная область или жанр являются всего лишь той или другой модной одеждой, прикрывающей более постоянную неприглядную суть.

Злой шутник найдет способ пошутить и без компьютера, а для молодого коммерсанта видеолента ни-

чем не хуже дискеты с компьютерными играми. Вы правы — были бы деньги, а повод злоупотребить ими всегда найдется.

Возможность застрять на элементарных умениях не может отменять необходимость изучения основ. Точно так же эти основы не могут быть компенсированы способностью работы с конечным программным продуктом. Вы сами говорите: «и немного программировать самому». Так вот, электронные таблицы удивительно удобны, пока Вы находитесь в пространстве заданных возможностей. Однако даже самая невинная попытка выйти за эти пределы превращает удобнейший инструмент в очень неуклюжую систему программирования. И если разматывать эту цепочку нарастающих проблем, Вы не сможете остановиться, пока не упретесь в необходимость элементарной алгоритмической нотации, поддерживающей принципы структурного программирования.

Что до «Ершолола», то мне лестна подобная персонификация. Однако справедливости ради я не могу считать алгоритмический язык Информатики-9 своим изобретением. Это

общее достояние, сложившееся за последние 20 лет и уже увековечившее себя в сотнях статей, десятках книг и ряде систем и технологий программирования. Единственное, за что я берусь отвечать, — это тезис, что эта сложившаяся система обозначений достойна занять свое место рядом с грамматикой родного языка, рядом с языком математических символов.

В заключение об «интуитивном программировании» по Г. Р. Грому. Здесь, мне кажется, надо различать три вещи: 1) интуитивный, эмпирический, «снизу вверх» способ постижения грамматики языка; 2) уровень искусства во владении языком программирования и 3) выбор наиболее подходящего языка для реализации «автоформализации профессиональных знаний». Думается, что уже само осознание этих различий и возможность спорить об одном, не затрагивая другого, позволит найти рациональный баланс энтузиазма и критики.

А. Ершов

рым повтoрам. Разумеется, это не исключает, а скорее предполагает, что в кулуарах семинара мы продолжим работу со всеми, у кого остались такого типа неясности, как и принято на всех наших встречах.

Вторая группа — это общие, риторические по стилю дискуссионные вопросы, посвященные некоторым философским, методологическим и педагогическим аспектам формирования критериев оценки профессионального уровня в той или иной предметной области, а также различного рода «метрикам» для соответствующего ранжирования профессиональных знаний, в том числе и по уже упомянутой в докладе шкале «относительной научности». Трудно предположить, что за одно заседание мы могли бы успеть здесь, хотя бы поставить все эти вопросы на обсуждение. Некоторые из них, имеющие ясно выраженную прагматическую основу, могут быть обсуждены на семинаре: «Технология автоформализации профессиональных знаний», который работает в Научном центре Пушино; другие, возможно, требуют внимания более специальной аудитории.

В наиболее общей форме ответ на многие из вопросов этой группы, как нам представляется, содержится в замечании А. С. Макаренко:

«Вот, что я вам скажу. Не приходило ли вам в голову, что высшая профессиональность состоит именно в том, чтобы уметь без вреда для дела стать иногда выше этого самого профессионального к нему отношения? И, будучи во всеоружии знания и понимания в своей специальности, суметь взглянуть на вещи прямо и просто, вооружившись одним только здравым смыслом? Вы не согласны?»

И, наконец, третья группа — это, на наш взгляд, наиболее «прицельные» из дискуссионных вопросов. Остановимся на них более подробно. Значит ли все выше сказанное, что традиционные методы формализации знаний утрачивают свои позиции?

С нашей точки зрения, как раз наоборот, рост важности систематически используемых строгих методов «полного решения» формально поставленных задач непосредственно следует из ожидаемого быстрого расширения области их приложений. Значительную роль играют при этом методы автоформализации профессиональных знаний, которые, как мы пытались показать, нередко «прокладывают дорогу» для последующего использования более строгих методов в решении той или иной впервые поставленной на ЭВМ прикладной задачи.

Наконец, хотелось бы в этом контексте еще раз подчеркнуть, что по центральной своей сути концепция автоформализации означает не отрицание полезности какого-либо из ра-

нее сложившихся методов формализации, а последовательное расширение области производственных приложений ЭВМ за те рамки, в которых традиционные методы давно и честно работают. По-существу, мы лишь отходим от сложившегося до сих пор «бинарного» принципа отбора прикладных задач: «все — или ничего», — который существенным образом ограничивал масштабы компьютеризации народного хозяйства.

Итак, есть области приложений, где традиционные методы позволяют получить доказательное «полное решение», но есть и такие, где методы традиционной математики пока не работают. Означает ли это последнее обстоятельство, что следует «декретно» приостановить процесс автоматизации в «трудных зонах», объявив их «донаучными», «недозревшими» и т. д.? Как заметил однажды Хевисайд, «мне не мешает испытывать удовольствие от вкусного ужина тот факт, что я не представляю себе точной картины процесса пищеварения».

МИКРОПРОЦЕССОРЫ В ПОЛИТЕХНИЧЕСКОМ

В 1986/87 учебном году журнал проведет в Политехническом музее два учебных цикла:

МикроЭВМ: основы применения

Цикл ориентирован на руководителей предприятий, профессорско-преподавательский состав вузов, школ, техникумов, а также на студентов и широкий круг специалистов различных отраслей народного хозяйства.

1. Персональные компьютеры: серийные изделия отечественной промышленности — 9.IX 1986 г.
2. Школьная информатика: второй звонок — 14.X 1986 г.
3. Компьютер в вузе: качественно новый учебный процесс — 11.XI 1986 г.
4. Автоматизация учреждений (совместный семинар журналов «ЭКО» и «Микропроцессорные средства и системы») — 9.XII 1986 г.
5. МикроЭВМ в системах автоматизации проектирования и производства — 13.I 1987 г.
6. Компьютерные игры — 10.II 1987 г.
7. Диалог человек — ЭВМ: вопросы психологии — 10.III 1987 г.
8. Микропроцессоры в массовых изделиях бытовой электроники — 14.IV 1987 г.
9. Компьютер и медицина — 12.V 1987 г.
10. Компьютер и музыкальное творчество — 9.VI 1987 г.

Микропроцессоры — рабочий инструмент инженера

Цикл ориентирован на разработчиков микропроцессорной техники, а

Иными словами, мы не обсуждаем полезность строительства автострад, а лишь обращаем внимание на то обстоятельство, что, несмотря на огромный и постоянно растущий грузооборот дорог с «твердым покрытием», их суммарная площадь составляет все еще лишь ничтожную часть земной поверхности. Поэтому, видимо, всегда, на любом уровне развития «дорожного строительства» будет оставаться актуальным вопрос, как быть «там, где кончается асфальт». Можно, конечно, ответить, оставаясь в рамках достаточно известной «пуристской» позиции: «Надо подождать! Если объект, о котором Вы беспокоитесь, действительно настолько важен, то в плановом порядке туда в свое время проложат асфальт...» Однако, известен и другой подход, согласно которому тропинки сначала протоптавывают, а потом асфальтируют.

Тем, кто сегодня протоптывает тропинки на месте будущих автострад большой науки, и был адресован этот доклад.

также инженеров других специальностей, которым необходимо работать с новой информационной техникой.

1. Обзор основных классов отечественных микропроцессоров и микроЭВМ — 23.IX 1986 г.
2. Аппаратная структура и программное обеспечение типового микропроцессорного комплекса — 28.X 1986 г.
3. Особенности схемотехнической реализации и методы отладки микропроцессорных систем управления — 25.XI 1986 г.
4. Устройства связи микроЭВМ с объектами автоматизации — 23.XII 1986 г.
5. Периферийное оборудование микроЭВМ: дисплеи, гибкие диски, электронный «квизидиск», ЦМД-память, клавиатура, печатающие устройства — 27.I 1987 г.
6. Аппаратно-программная реализация многомашиных комплексов и локальных вычислительных сетей — 24.II 1987 г.
7. Операционные системы микроЭВМ — 24.III 1987 г.
8. Программируемые логические матрицы: основные классы, технические характеристики, типовые примеры применения — 28.IV 1987 г.
9. Техника работы с ПЗУ: физические основы и методы программирования, схемотехника универсальных и модульных программаторов — 26.V 1987 г.
10. Тенденции развития индустрии ЭВМ — 23.VI 1987 г.

Справки о семинарах можно получить по телефону 923-00-19 у методиста Политехнического музея Ермолаевой Татьяны Юрьевны. Абоненты могут приобрести в кассе Политехнического музея (подъезд № 9).

Хочу обратить Ваше внимание на опасность массового распространения «компьютерной полуграмотности». Я имею в виду не недостаточное качество обучения на сегодня, я знаю Вашу точку зрения, что важно хотя бы начать. Вопрос в том, что многие готовы были бы считать вопрос о грамотности решенным (и «закрытым») в результате обучения чему-нибудь вроде Бейсика. И вопрос не в том, что Бейсик плох (или, может быть, хорош), а в том, что это совсем не то умение, которое поможет массовому специалисту, нуждающемуся в использовании вычислительных средств. Не потому, что не нужно в школе знать какой-то язык типа «мини-языка» Информатика-9 (не знаю, как его называть, но свято место пусто не бывает, и злые языки уже прозвали его «Ершолом»). Только этот язык (как и формируемое на его основе представление об алгоритме), скорее, относится к математическому курсу, а не к информатике (аналогично знанию прогрессий или символа Σ).

Всерьез специалисту нужно не кропать для себя мелкие самостоятельные программки (их возможности, особенно у непрофессионала, весьма ограничены), а уметь обращаться к готовым средствам, дополняя их понемногу собственным программированием. Но против чего я больше всего здесь возражаю, это против ориентации на «инстинктивное» программирование, по Г. Р. Грому. Думаю, что в сложном случае человек при таком подходе просто не справится с задачей и махнет рукой на программирование вообще.

Появилась и другая серьезная опасность. Старшие школьники и студенты, «дорвавшиеся» до машины и предоставленные самим себе, замыкаются в примитивном круге задач, превращаются в «фанатиков» (термин моих студентов, наблюдавших это явление в свою бытность в школе-интернате). Их больше всего начинает интересовать хитроумное использование каких-либо малоизвестных возможностей машины или операционной

системы, возможность обойти разные запреты, недобро подшутить над соседом. Я уж не говорю о пролезании на машину с черного хода или ночных бдениях. Замкнувшись во всем этом, иные утрачивают способность перейти к более серьезным задачам, вылетают с факультета, а то и попадают в больницу. Или такой «пострел» уже на первом курсе начинает наводить справки, у кого бы купить за наличные бобину магнитной ленты (краденую, естественно). О заграничной породе «хакеров» (hackers) Вы, вероятно, лучше меня знаете. Это явление, может быть, не специфично для программирования: В. А. Залгаллер, учивший нас в университете геометрии (и заодно уму-разуму), рассказывал об опасности для способного школьника или студента застрять в круге задач элементарной математики.

И не стоит отвечать, что все эти ребята со временем «перебесятся», зато приобретут необходимую в других делах виртуозность. Я никак не против игры и не против виртуозности, но если мы не поведем этих людей вперед, ставя перед ними такие задачи, по сравнению с которыми всякие машинные трючки покажутся им неинтересными, то само это не пройдет. И не надо ссылаться на наше поколение, успешно миновавшее эту фанатизацию. Вот аналогия: моего сына кое-кто упрекает в злоупотреблении сладкой пищей. Я попытался вспомнить, почему мне такое не повредило. И вспомнил: денег не хватало.

Здесь о молодежи я сказал то же, что о взрослых, ликвидирующих «вторую безграмотность». Нельзя останавливаться на элементарных навыках программирования, а может быть, и вообще не стоит делать на них упор.

Доктор физ.-мат. наук Г. С. Цейтин,
математико-механический факультет
Ленинградского государственного университета

Григорий Самуилович!

Разделяя в целом Ваши предостережения об опасности «компьютерной полуграмотности», хотел бы сделать ряд замечаний в защиту различных сторон компьютерного всеобщего.

Мы воспринимаем термин «массовая культура» всегда с некоторыми оговорками, не упуская ни одного повода поговорить об ее издержках. Однако уже в силу объективных законов социальной психологии мы не можем от любого крупномасштабного явления оставить только его одну, «лучшую» сторону в надежде избавиться от другой, «худшей».

Полуграмотность, бескультурье, злоупотребление, сектантский фанатизм — это, к сожалению, универсальные стороны человеческого бытия, для которых предметная область или жанр являются всего лишь той или другой модной одеждой, прикрывающей более постоянную неприглядную суть.

Злой шутник найдет способ пошутить и без компьютера, а для молодого коммерсанта видеолента ни-

чем не хуже дискеты с компьютерными играми. Вы правы — были бы деньги, а повод злоупотребить ими всегда найдется.

Возможность застрять на элементарных умениях не может отменять необходимость изучения основ. Точно так же эти основы не могут быть компенсированы способностью работы с конечным программным продуктом. Вы сами говорите: «и немного программировать самому». Так вот, электронные таблицы удивительно удобны, пока Вы находитесь в пространстве заданных возможностей. Однако даже самая невинная попытка выйти за эти пределы превращает удобнейший инструмент в очень неуклюжую систему программирования. И если разматывать эту цепочку нарастающих проблем, Вы не сможете остановиться, пока не упретесь в необходимость элементарной алгоритмической нотации, поддерживающей принципы структурного программирования.

Что до «Ершолола», то мне лестна подобная персонификация. Однако справедливости ради я не могу считать алгоритмический язык Информатика-9 своим изобретением. Это

общее достояние, сложившееся за последние 20 лет и уже увековечившее себя в сотнях статей, десятках книг и ряде систем и технологий программирования. Единственное, за что я берусь отвечать, — это тезис, что эта сложившаяся система обозначений достойна занять свое место рядом с грамматикой родного языка, рядом с языком математических символов.

В заключение об «интуитивном программировании» по Г. Р. Грому. Здесь, мне кажется, надо различать три вещи: 1) интуитивный, эмпирический, «снизу вверх» способ постижения грамматики языка; 2) уровень искусства во владении языком программирования и 3) выбор наиболее подходящего языка для реализации «автоформализации профессиональных знаний». Думается, что уже само осознание этих различий и возможность спорить об одном, не затрагивая другого, позволит найти рациональный баланс энтузиазма и критики.

А. Ершов