

# Научная фантастика и научная реальность в информатике

Эдсгер Вибе Дейкстра

Как среди практиков информатики, так и среди широкой публики бытует масса недоразумений касательно информатики, и цель данной беседы — прояснить эти недоразумения, поскольку они вредят нам всеми мыслимыми способами.

С одной стороны, достижения информатики досадно игнорируются. Многие проекты,

в которых осознанное применение информатики необходимо, выполняются в совершенно ненаучной манере, как будто информатики не существует вовсе. Это не может не вызывать сожаления, поскольку приводит ко многим дорогостоящим ошибкам, которых можно было бы избежать, а информатика не получает должного признания своих достижений.

С другой стороны, ожидания относительно того, что информатика может нам дать, зачастую совершенно нереальны: ожидается как минимум череда чудес целыми дюжинами. И это тоже весьма досадно, поскольку ведёт к неоправданным надеждам и нереальным планам, а когда эти чудесные планы проваливаются с треском, информатика теряет доверия как ещё одна из форм шарлатанства.

Если желаете, можете рассматривать этот доклад как попытку изменить то, что ка-

жется сейчас основным затруднением для учёного-информатика: прежде всего ему не дозволено иметь какого-либо влияния, а затем на него валят все шишки за любые неудачи.

На первый взгляд кажется, что разъяснение квинтэссенции науки и затем получение выводов о свойственной ей роли — чисто техническая задача, но, к сожалению, это не так. Проблема состоит в том, что это противоречит интересам промышленности и чересчур неудобно для многих, чтобы быть услышанным. В прошлом от этого бы просто избавились. Это вызывало дискомфорт на некоторое время, после чего публика осмеивала это, и всё возвращалось на круги своя, а провозвестника идеи терпели, как терпят выходки шута. Но состояние дел изменилось, интересы промышленников стали обширнее и разнообразнее.

разнее, и всё больше и больше людей начинают понимать, что уже становится не до смеха. По мере того как разумное и бесстрастное обсуждение этих вопросов становилось всё более неотложным, становилось всё труднее и труднее говорить о них. За техническими деталями я отправляю вас к судьбе Галилео Галилея.

Ранее я говорил, что достижения информатики, к сожалению, игнорируются. Что же, если бы это было именно так, не было бы повода для тревоги: разрыв между полезными достижениями в лаборатории и их практическим применением неизбежен. В нашем случае, однако, есть повод бить тревогу, поскольку на протяжении последних десятилетий разрыв между информатикой и практическим использованием компьютеров только расширился. В то время как информатика де-

лала большие шаги к превращению в строгую, точную науку, компьютерная практика в основном пребывала в застое. Я не преувеличиваю: физики до сих пор убеждены, что FORTRAN — последнее слово в информатике, химики продолжают использовать BASIC, и COBOL стал для бакалавра тем же, что и APL для инженера-электронщика. Человеческая склонность привязываться к источнику своих бед зарекомендовала себя как важный стабилизирующий фактор во многих браках и религиях, однако из-за своей болезненной склонности к неадекватным инструментам эти дисциплины переступили черту, за которой им уже не помочь.

Эти наблюдения обычно относятся к «случайному пользователю», но в последнее время проблема усилилась из-за того, что она повлияла на многие университетские учебные

курсы таким образом, что вместо первоклассных учёных они теперь готовят третьесортных программистов.

Что же, вы можете возразить, что эти люди сами себя считают не программистами, а физиками, химиками и т.д., но и в профессиональной области картина столь же безрадостна. Мы все слышали о «Шаттле», который не был запущен из-за ошибки в программном обеспечении синхронизации, поэтому я должен привести вам другой пример. Недавно Британская железная дорога установила свою первую компьютерную систему сигнализации на одной из веток, и они разрекламировали в надежде продать её другим железнодорожным компаниям, что во избежание риска от использования компилятора в целях безопасности система была написана в машинных кодах. Очевидно, даже не ставилось

под сомнение, что промышленные компиляторы сыграют с вами злую шутку. Другая область, полная страшных историй, — разработка VLSI (СВИС, сверхбольшие интегральные схемы — прим. перев.). В арсенале разработчиков среди прочих инструментов имеется программа — очень дорогая в эксплуатации, — которая пытается восстановить схему по фотошаблону, поскольку программа, производящая фотошаблон, ненадёжна. Но алхимия коммуникационных протоколов превосходит их всех, вместе взятых; первоначально разработанные телекоммуникационными инженерами для компенсации случайных аппаратных сбоев, они стали настолько громоздкими, что вызывают гораздо больше ошибок передачи, чем те, которые они должны были бы исправлять, и проверка их реализации стала сложной задачей: распутывание этого

клубка может стать темой диссертации. И всех этих несуразностей можно было бы избежать, поскольку информатика начала углублённое и успешное изучение проблем компиляции и синхронизации двадцать с лишним лет назад. Слишком много для разрыва между наукой и практикой.

Мы уже видели, что программы учебных курсов меняются под действием необоснованного предположения, что в наши дни использование компьютера является основным, чтобы стать хорошим учёным. Сейчас это представляется весьма странным, поскольку этому выводу противоречит факт, что существует большое количество чрезвычайно успешных учёных, которые никогда не использовали компьютер, так как в их время компьютеров попросту не было. Верно, что микрокомпьютеры продаются столь же широко,



как и энциклопедии, компрометируя родителей будущим их чад. А если вы читаете между строк, призыв звучит ещё более заманчиво: дайте вашему ребёнку наш домашний компьютер, и он вырастет гением. Но нам не следует винить промышленность за это недоразумение, как бы настойчиво она не пыталась запихнуть свои леденцы в наши глотки: в подобных рекламных ухищрениях они лишь используют нашу скрытую, но не покидающую нас мечту о «философском камне», способном делать золото, и «Эликсире», дарующем вечные молодость и здоровье. Промышленность ничем не хуже обычного проходимца.

Насколько мне известно, не промышленность, а общество придумало термин «компьютерная грамотность». В самом начале игры в человеческое сознание прочно вбили мысль, что компьютеры появились, чтобы

спасти нашу экономику (а позже — также и экономику стран третьего мира). Некоторое время никто не знал, каким именно образом, но, к счастью, кого-то посетила блестящая идея, что информация — это «ресурс», и это решило проблему, тем более что с этого момента, впервые в истории человеческого разума, мы наконец-то получили ценный ресурс, запасы которого неисчерпаемы. Затем люди вспомнили, как нефть принесла сказочные богатства немногим счастливым; на этот раз подобного не должно было произойти, поэтому для решения этой проблемы была изобретена «компьютерная грамотность» для миллионов. Что этот термин должен означать, до сих пор является открытым вопросом, однако это не мешает нашим оракулам продолжать грезить о постиндустриальном обществе, в котором люди, когда им надо-

ест наслаждаться телеконференциями, будут посвящать свой обширный досуг созидательным видеоиграм.

Нет ничего нового в том, что люди ожидают рая на земле в ближайшем будущем; новым, однако, является то, что это будет создано при помощи микроэлектроники, а не милостью божьей. Я не преувеличиваю: ожидания эти столь нереальны, что вещи, которые следовало бы разумно обсудить, всё чаще преподносятся так, словно люди участвуют в странном ритуале, истово повторяя слова жрецов на языке, которого они не понимают. Позвольте привести лишь несколько свежих примеров.

Директор крупной исследовательской лаборатории во время интервью по поводу его ухода на пенсию преподносил достоинства домашнего компьютера, подчёркивая, что он

позволит людям «упорядочить их творческие мысли». Только подумайте на минуту, какая бессмыслица! Между нами, когда вас в последний раз посещала творческая мысль? Пару лет назад? Неплохо. Гораздо больше смысла было бы, если бы он сказал «привести в порядок вашу коллекцию марок». Когда Бертран Рассел обронил своё знаменитое изречение «Многие люди предпочтут скорее умереть, чём подумать. Это в самом деле так», он высказал больше реализма. Лично я сделал для себя вывод, что этот директор ушёл на пенсию не слишком рано.

На международной конференции, посвящённой компьютерам в образовании, все сошлись на мысли, что маленьких детей следует знакомить с компьютером при первой возможности. Возникла некоторая озабоченность, что столь интенсивная механизация

образовательного процесса могла бы повредить их эмоциональному развитию и образованию навыков общения, пока кто-то не заметил, что «разговор с программой не причинит вреда, поскольку, будучи продуктом человеческой мысли, программа обладает в основном человеческими чертами». Теперь я задумываюсь, не было ли окончательное решение Гитлера результатом человеческой мысли?

На последней конференции относительно осуществимости или неосуществимости разработки программного обеспечения для проекта «звёздных войн» одним из аргументов в пользу осуществимости, как мне сказали, было замечание, что сегодня почти все банковские операции успешно компьютеризованы, не так ли? Ну что ж, по правде говоря, это не так: не далее как в 1986 году попытка выплачивать зарплату электронными средства-

ми в крупной (компьютерной!) компании провалилась. Но если бы даже это было и так, это ложный аргумент: эти два проекта настолько различны, что аналогия на несколько порядков величины мельче, чем нужно для принятия её во внимание. Хуже всего, я считаю, то, что этот аргумент был выдвинут человеком, занимающим видное положение в академической среде.

В шестидесятых годах сэр Герман Бонди написал статью о том, кому следует заниматься академическими исследованиями. Вопрос был поднят весьма важный. До второй мировой войны учёных было пренебрежимо мало, но после признания того, что война была выиграна наукой и технологиями, академические учреждения начали получать поддержку. Неудивительно, что Бонди пришёл к выводу, что многим из его современ-

ников, привлечённых к научным исследованиям, лучше было бы подыскать себе другое занятие. Поскольку взрывное развитие университетов было напрямую связано с ожиданием большой отдачи от науки, Бонди также посвятил раздел своей статьи обсуждению пользы от науки в целом. Его вывод отрезвляет.

Он указывает, что среди задач, поступающих в университеты извне, из «реального мира», около 80% тривиальны и около 20% явно неразрешимы, и что академические исследования потенциально могут повлиять на тонкий граничный слой между ними, так как только там знания, талант и упорный труд могут достичь чего-то, что не может быть достигнуто другими средствами. В нынешней научно-технической эйфории столь отрезвляющее предупреждение более чем необходимо;

оно, впрочем, не было ни услышано, ни замечено, но это уже другая история.

По несчастному историческому совпадению, компьютеры появились как раз в десятилетия безграничной веры в всемогущество науки и технологии. Тогда как более устоявшиеся науки обзавелись корнями и традициями в более благоразумные времена, мы ввязались в информатику, когда наши ожидания ничем не сдерживались. С исторической точки зрения этот наивный оптимизм вполне понятен. И этому даже можно найти кое-какие оправдания. Видите ли, Бонди физик, и его мнение 80/20% обусловлено его знанием, каким образом естественные науки справляются со сложностями, с которыми мы сталкиваемся в окружающем нас мире. Напротив, информатика имеет дело с миром артефактов, в котором сложность имеет рукотворное



происхождение. В этом заключается большая разница между ними, и вполне возможно, что в области информатики «пограничный слой» Бонди вовсе не так уж исчезающе тонок. Но чтобы сделать грамотный вывод о том, что данный слой мог бы и должен был бы содержать, нам придётся понять, как возникла и как развивалась информатика.

Впрочем, перед тем, как обратиться к информатике как таковой, я должен разделаться ещё с одной мечтой, которой часто грезили в эти десятилетия, а именно — что возможно «планировать» исследования таким образом, чтобы в должное время производить то, что требует мир. Наука не может работать в таком режиме (что, кстати говоря, не так уж плохо, поскольку зачастую существует коренное различие между тем, что мир требует, и тем, что ему в действительности нужно).

Е.Т. Белл справедливо хвалил некоторых руководителей за то, что они были «достаточно дальновидны, чтобы понять, что самый простой путь избавить математику от математики — это оплачивать его содержание», и это справедливо и для других наук. Никогда ещё не наблюдался значительный прогресс в науке из-за того, что какому-то благодетелю понадобился результат. Существенный прогресс в науке возникает лишь тогда, когда после поисков и размышлений образованный и оригинальный ум приходит к выводу, что что-то загадочное наконец-то созрело для понимания или что-то очень трудное может теперь быть сделано. Успешный научный поиск — это искусство делать действительно возможное, и следовательно, развитие науки гораздо лучше рассматривать как автономный процесс со своими собственными правилами, а не как

плановую деятельность с заранее установленными внешними целями. (И с этим парадоксом сталкиваются все директора промышленных исследовательских лабораторий: привлекая нужных людей, они не могут служить своей компании лучше, чем просто оставив их в покое).

Сторонники междисциплинарных исследований, кажется, иногда верят в то, что границы между различными науками — не более, чем каприз истории. Но пути, которыми научное знание расходилось по разным дисциплинам, вовсе не случайны: то, что может образовать область жизнеспособной научной дисциплины, должно иметь количественные и качественные ограничения.

Среди количественных ограничений я упомяну, что область должна быть достаточно мала, так чтобы основные положения уме-

щались в одной человеческой голове; с другой стороны, она должна быть достаточно обширна, чтобы предоставлять пищу для размышлений хотя бы на некоторое время.

Среди качественных ограничений я упомяну, что с одной стороны эти проблемы должны быть достаточно независимы от остальных, чтобы их можно было изучать изолированно от других, тогда как с другой стороны область должна обладать внутренней согласованностью.

Последнее требование полностью проясняет, каким образом раньше отделения информатики предшествовали самой информатике: поначалу они представляли собой не более чем малопонятный коктейль всевозможных дисциплин, имеющих отношение к компьютерам, которые посчастливилось наскреести по университету, например: немного элек-

троники, немного численного анализа, немного статистики и экономики, немного бизнес-администрирования и в США в довершение всего — немного искусственного интеллекта. Коктейль, составленный как попало, не может быть хорош на вкус. Образование связей — одна из первостепенных задач для тех, кто пробивает нишу, в которой многообещающая наука информатика станет жизнеспособной.

Ради этой согласованности они оставили все специфические области потенциальных компьютерных приложений и попытались сконцентрироваться на том, что все эти приложения должны были иметь общего; они сделали это ради согласованности и общности, которая необходима для надёжности.

Другой аспект этой надёжности — сознательная попытка избежать подготовки учёных в течение половины от пятилетнего сро-

ка. Это включает, в частности, воздержание от всего, что только имеет отношение к компьютерам, доступным нынче на рынке. Например, то, как обходиться с идиосинкразией OS/360, рассматривалось как временная цель, не относящаяся к предмету информатики.

В общем: ниша была подготовлена вне специфических приложений и вне специфических машин. С течением времени в том же духе продолжается уклонение от особенностей специфических языков программирования и операционных систем. Сначала это делалось для защиты зарождающейся науки от изменчивости рыночных продуктов, а когда некоторые из этих продуктов стали стандартом де-факто, это стало делаться для защиты расцветающей науки от рыночного застоя. Независимо от того, выражать ли сожаление

или восторг по поводу этого разделения, я хочу, чтобы вы поняли, что для становления информатики как жизнеспособной науки это разделение было и остаётся *conditio sine qua non*.

Что же информатика делала в этой блистательной изоляции? Точнее говоря, что она делала, не теряя претензий на прикладную значимость? Что же, сделала она немало; фактически, куда больше, чем я могу разъяснить в рамках данной лекции, но я могу хотя бы передать вам общие моменты.

В шестидесятые годы она разработала теорию синтаксического разбора, необходимую для поднятия уровня компиляторов выше уровня поделок, напичканных ошибками, и превратив её в предмет, пригодный для обучения. Это было главное достижение: я, например, помню весьма отчётливо, как в 1962

те из нас, кто действительно написал компилятор, выглядели в глазах остальных как некие полубоги. В связи с этим достижением я бы хотел подчеркнуть, что этого никогда бы не произошло, если бы мы со временем не научились давать формальное определение синтаксиса компилируемого языка: без этого формального определения слишком сложно было бы определить существование проблемы компиляции. Теория конечных автоматов и теория сложности были разработаны, чтобы задать основные количественные границы того, что в принципе может быть вычислено; опять же, в основе этих теорий лежит очень формальный постулат относительно того, что есть вычисление, постулат, без которого эти теории не могут существовать. Для разработки операционных систем проблема синхронизации процессов была поставлена и решена, и



первые теоремы об отсутствии «смертельных объятий» были доказаны; также формальное определение явления, интуитивно известного как «смертельное объятие», было первой предпосылкой этого достижения.

В семидесятые годы центр внимания сместился от синтаксиса к семантике, сначала к детерминированным последовательным программам, но вскоре впоследствии охватил также недетерминированность и параллельность. Я не буду подробно описывать различные отрасли: они простираются от модели типизированного лямбда-счисления до разработки преобразований программ с сохранением семантики. В этом десятилетии программы стали самостоятельными математическими объектами. Кратчайший способ уловить изменение направления внимания — это, пожалуй, отметить, что если раньше задачей

программ было управлять поведением машины, то теперь задачей машины стало выполнение наших программ. Верификация и разработка программ развились в разделы формальной математики до такой степени, что теперь уже не считается безответственностью опубликовать программу, не испробовав её на компьютере.

Что ж, это далеко не полный обзор того, как информатика стала наукой, и я приношу свои извинения всем неупомянутым мной, кто внёс свой вклад в её становление. Но я надеюсь, что он всё же достаточно полон, чтобы донести до вас аромат квинтэссенции дисциплины. Она стала замечательной дисциплиной, поскольку разделение между «чистой» и «прикладной», столь традиционное для многих других дисциплин, совершенно поблёкло и существенно утратило своё значение. Рос-

кошь работы в окружении, в котором различие между чистой и прикладной наукой лишено смысла, — это, пожалуй, ещё одно признание факта, что компьютер общего назначения действительно заслуживает эпитета «общего назначения».

Она обладает всей пикантностью чистой математики, будучи более формальной, чем многие другие отрасли математики. Она не может избежать такой формальности, поскольку любой язык программирования, будучи интерпретируемым механически, представляет своего рода формальную систему. В то же время она обладает всей прелестью прикладной математики, поскольку огромная мощность современных компьютеров даёт такие возможности для создания хаоса, что её методы необходимы, если мы не намерены попасться в ловушку сложности, которую са-

ми же и создали.

Научиться не попадаться в собственноручно произведённую ловушку сложности, сохранять вещи достаточно простыми и научиться достаточно эффективно мыслить о своих разработках — вот центральная задача информатики. Это также было осознано больше десятка лет назад, когда «разделение задач» стало находкой программной терминологии.

Заметьте, пожалуйста, что путь, которым информатика пробивала свою нишу, был сам по себе примером успешного «разделения проблем». Каждый раз, когда мы вкладываем уйму умственной энергии в тщательную разработку любой дискретной системы, мы делаем это не просто для удовольствия: мы всегда надеемся, что результат наших усилий будет использован на благо другим. Мы

надеемся, что он будет удовлетворять нужды, соответствовать ожиданиям и доставлять удовольствие своим пользователям. В доначный период разработки систем неформальное понятие «удовлетворения пользователя» было единственным приемлемым критерием качества программного обеспечения.

Недостаток «удовлетворения пользователя» как критерия качества состоит в том, что это не техническое понятие: он не задаёт технического направления разработчику и, кроме того, может быть достигнут другими средствами, помимо технических, например, агрессивной рекламой или промывкой мозгов. К науке это не имеет ни малейшего отношения. Задачи должны быть разделены, и тут на сцене появляются функциональные спецификации.

Роль формальной функциональной спе-

цификации — просто служить логическим барьером между двумя совершенно разными проблемами, известными как «проблема удовлетворённости» и «проблема корректности». Проблема удовлетворённости касается вопроса, соответствует ли система, отвечающая таким-то и таким-то формальным спецификациям, вашим ожиданиям и надеждам. Проблема корректности касается вопроса, соответствует ли данная разработка таким-то и таким-то формальным функциональным спецификациям.

Логический барьер был необходим, чтобы поставить проблему корректности перед информатикой: он изолирует уютную нишу информатики от проблемы удовлетворённости, решению которой наука мало чем способна помочь. Заметьте, пожалуйста, что я не утверждаю, что одна из проблем важнее дру-

гой; в конце концов, цепь не прочнее самого слабого звена. Однако я утверждаю, что проблема корректности представляет ту самую часть, которую нам удалось втиснуть в «тонкий пограничный слой» Бонди, где разумное применение научной мысли может принести пользу, тогда как неформализованная проблема удовлетворённости в действительности лежит на пределах компетенции науки.

У вас могут быть самые разные проблемы, начиная с опасного перекрёстка и заканчивая такими, которые угрожают самому существованию целых поколений. Однако наука никогда не решает ваши проблемы, она решает лишь свои собственные, и заключение, примете ли вы решение формальной научной проблемы как таковой в качестве приемлемого решения своей задачи, лежит целиком на вас. Другими словами: наука ни-

когда не предлагает моделей реальности, она лишь строит свою теорию, и вопрос, примете ли вы своё восприятие действительности в качестве достаточно достоверной модели для этой теории, полностью ваша проблема. Достоверность и реалистичность больше не являются научными понятиями, и учёный уступает право разглагольствования о них философам, пророкам и поэтам.

С этой точки зрения роль науки довольно ограничена, на самом деле до такой неутешительной степени, что многие предпочитают закрывать глаза на ограниченность науки. Акцентировать внимание на этих ограничениях не принято в научной среде, поскольку помимо прочего это вызывает вопрос, а зачем тогда общество должно терпеть учёных. Это не шутка: все мы знаем, что если бы сегодня общество вдруг решило изгнать своих



учёных, это было бы не впервые.

И теперь, когда наука «вышла в народ», так сказать, она также не столь популярна среди публики в целом. Люди всегда питали противоречивые чувства по отношению к технологии, и с чем более мощными технологиями они сталкиваются, тем драматичнее становится двойственность этого отношения. Они чувствуют угрозу со стороны технологии сильнее, чем когда-либо прежде; в то же время их надежда на спасительную мощь науки и технологии всё больше крепнет. В старые добрые времена традиционного шаманства от него требовались только снадобья от всех недугов, Эликсир для вечной юности и Философский Камень для сотворения золота. Так было в старые добры времена, когда поиск Философского Камня только начинался.

В наши же дни передовой электроники

Философский Камень обзавёлся совершенно новыми измерениями. Достижения в традиционном стиле сотворения золота благодаря робототехнике постепенно во всех странах приведут к положительному балансу в торговле. Они решат проблемы производства и безработицы. Они дадут правительству силы победить преступность и коррупцию, тогда как вездесущие микрокомпьютеры станут на страже демократии. Обучающие машины обновят образовательный процесс, в то время как калькуляторы, автоматические корректоры правописания и в целом мыслящие машины сделают большую часть обучения излишней. Все государственные секреты будут абсолютно надёжно защищены неприступным шифрованием; мощные схемы декодирования позволят нам взламывать любые коды. Оружейные и защитные системы будут одинаково-

во эффективны. И, что важнее всего, если мы не знаем, что нам делать, мы получим «поддержку принятия решений», наше руководство будет информировано, а наша информация — управляема, нашим разработкам будет оказываться столь же значительная помощь, что и усилению наших мыслительных способностей, и безо всякой специальной подготовки любой, в самом деле любой, даже руководство и генералы, будут иметь под рукой все нужные им знания экспертов. Замечательное новшество сегодняшнего Философского Камня состоит в том, что вы можете перевалить на него свою ответственность.

Слишком много для вопиющей бессмыслицы. Теперь можно было бы поспорить, что в своей несклонности к компромиссам я занял непримиримую позицию; я знаю, что немало из моих благоразумных и уважаемых коллег

критикуют бы пределы, которые я обрисовал им для информатики непрактично узкими. Они указывают на всевозможные системы, обладающие большой потенциальной полезностью, лежащие за пределами очерченных мной строгих границ. Они возражают, прежде всего, что нынче такие системы разрабатываются без особых формальных функциональных спецификаций, как бы нам этого ни хотелось. Один из примеров предоставляется библиотекой расчётных программ, для которой не границы применимости, ни точность результатов не были чётко установлены. Другим примером могла бы быть система оптического распознавания символов для использования в сортировке почты, по крайней мере будучи поставленной без точного определения, какие символы будут безошибочно распознаваться. Верно, однако позвольте мне

отметить несколько моментов.

Прежде всего, позвольте напомнить, что я отказался разделять проблемы удовлетворённости и корректности по их относительной важности, другими словами, что нет никакой внутренней порочности в ненаучном проекте: он всего лишь ненаучен. Во-вторых, существуют проекты, для которых очевидно, что ниша информатики не подходит; если это тот самый случай, информатика не в состоянии ничем им помочь, и лучше ей не вмешиваться. Почему компьютерные приложения, которым наука не способна помочь, должны быть отброшены? Я не думаю, что нам следует беспокоиться об этом. Я твёрдо убеждён в том, что мы сильно недооцениваем культурное значение компьютеров, судя о них в первую очередь как об инструментах, поскольку я ожидаю от них гораздо большего

влияния на их способность интеллектуально-го вызова.

Моё растущее беспокойство, однако, вызывает то, что подобные проекты используют лишь часть компьютерных характеристик, и то, что они ведут к достаточно неспецифицированным продуктам, в которых исчезают другие характеристики компьютера. Расчётные программы используют способность компьютера к перемалыванию чисел, оптическое распознавание символов использует ёмкость памяти и гибкость, но в обоих случаях конечный продукт больше не является автоматом с точно установленными свойствами. Следовательно, результирующий продукт уже не является машиной в смысле информатики, и его использование сродни занятиям математикой без аксиом. Подобные расчётные программы могут использоваться лишь в кон-

тексте, в котором результат не имеет столь большого значения, либо аналитик, использующий их, располагает другими средствами для проверки ответа; подобная система оптического распознавания может применяться лишь в обстоятельствах, в которых не имеет большого значения, если, скажем, часть писем сначала отправится в неверном направлении. Мы больше не можем полагаться на такие системы, как если бы они были автоматами, и нам придётся удалять их из замкнутых циклов.

Для многих такое заключение неприемлемо, и как результат существует школа мышления — или, если вам так больше нравится, школа недомыслия, — которая утверждает, что ситуация не столь серьёзна, что нам не следует быть столь строгими, что инженеры всегда допускали периодические сбои своих

компонентов, что стремление к совершенству быстро начинает препятствовать производительности, и что лучше бы нам научиться жить в реальном мире с системами и подсистемами, которые обычно делают то, что мы от них ожидаем. Это соблазнительное предложение. Разве не замечательно — достичь превосходства без гонки за совершенством? (Тем более что в крайне популистском обществе последнее социально неприемлемо). Такова позиция, ныне принятая под знаменем «программной инженерии».

Однако то, что сторонники называют «прогрессом программной инженерии», страдает несколькими непреодолимыми противоречиями.

Одно из них — это то, что стремление к совершенству находится в противоречии с производительностью в том смысле, что сделает



разработку программного обеспечения чересчур дорогостоящей. Но в чём основная причина заоблачных цен на неё? Дороже всего обходится как по людским ресурсам, так и по непредвиденным задержкам отладка, и можно немало сэкономить, вложив больше средств в предотвращение будущих ошибок, поставив разработку на первое место. Поскольку ошибки обходятся столь дорого, в конечном счёте высококачественный дизайн обходится куда дешевле. Другая важная причина состоит в том, что многие системы построены на зыбком фундаменте в том смысле, что базовое программное обеспечение в виде операционных систем и компиляторов слишком неустойчивы, поэтому каждый новый релиз этого базового программного обеспечения требует возможно дорогой адаптации базирующейся на нём прикладной части. Наконец,

многие инструменты, с которыми собирается работать программист, настолько плохо документированы, что вынуждают его выяснять экспериментальным путём, чем же они могут оказаться для него полезными. Поскольку эти эксперименты могут быть весьма дорогостоящими и длительными, бедный программист оказывается в поистине незавидном положении, поскольку вынужден полагаться лишь на свои догадки. Так что вы видите здесь три основных источника роста цены, след от которых ведёт к чьему-то предположению, что стремление к совершенству противоречит производительности!

Другое противоречие в развитии инженерии программного обеспечения всплывает, как только инженер-информатик задаётся вопросом, каким образом стать лучше, как искусство, мастерство или практика инженерии

программного обеспечения могут быть улучшены. Он немедленно обнаружит, что вынужден обратиться к дисциплине, которую отверг. Приняв в качестве хартии «Как программировать, если не можешь этого делать», инженерия программного обеспечения загнала себя в неприемлемую позицию, в позицию, которой информатика может избежать, лишь отказавшись от компромисса, придерживаясь своей собственной формальной дисциплины и не прикидываясь чем-то большим.

Чуть раньше я упомянул плохую документацию системы как внутреннее ограничение надёжности, с которой система может быть использована механически в более широком контексте. Теперь самое время указать, что привлечение технического писателя редко является выходом из положения; в сущности, это не более как признание того, что разра-

ботчики системы в некотором роде функционально безграмотны. Обычно даже целая армия технических писателей не может справиться с задачей, поскольку система становится столь сложной, что не поддаётся точному описанию.

Выдающийся пример этого явления недавно продемонстрировала Ada. Если Ada собирается выдать стандарт, желательно, чтобы он был недвусмысленно документирован. По меньшей мере две группы попытались сделать это; в результате обе выдали около 600 страниц формального текста. Это гораздо больше, чем необходимо, чтобы удостовериться в невозможности хотя бы твёрдо установить, что оба документа определяют один и тот же язык. Ошибка очевидной неуправляемости этих двух документов кроется ни в двух груп-

пах, составивших их, ни в принятом ими формализме, а лишь в самом языке: сами не обеспечив формального определения, могут ли его разработчики скрыть, что они предлагают неуправляемого монстра. То, что Ada уменьшит проблемы программирования и увеличит надёжность наших разработок до приемлемых границ, — это лишь одна из тех сказок, в которые могут поверить только люди с военным образованием. Самое лучшее, что я могу сказать об этом, — это упорные слухи, что даже военного образования недостаточно, чтобы поддерживать веру в этот Философский Камень. Я упомянул про Ada, поскольку это замечательный пример того, на что я указывал в начале доклада: её принятие — это политический процесс, в котором информатике, предостережения которой рассматривались как досадная по-

меха, не было дозволено оказывать никакого влияния. Следовательно, даже простое согласие с чьим-то обоснованным сомнением на этот счёт, даже без вмешательства в контроль и намерения, становится действием с политическим душком.

Я ещё не говорил об искусственном интеллекте. Что ж, этот предмет связан с другими политическими осложнениями, поскольку он стал частью заокеанской полемики: этот вопрос никогда не поднимался в Европе. В течение первых двух послевоенных десятилетий этому существовало простое финансовое объяснение. Искусственный интеллект был дорог, а Европа была бедна; к тому же искусственный интеллект финансировался почти исключительно министерством обороны, которое направило свои усилия на субсидирование — не могу сказать «поддержку» — амери-

канских исследований. Но одна лишь финансовая сторона не объясняет всего, поскольку, когда Европа стала достаточно богатой, чтобы финансировать собственные исследования в области искусственного интеллекта, об этом по-прежнему не велось и речи. На самом деле подобная участь постигла и другие отрасли науки программирования.

Моё заключение таково: это лишь один из аспектов намного более существенных культурных различий. Европейский разум поддерживает большее различие между Человеком и Машиной и ждёт меньшего от обоих. Он менее склонен описывать человеческую психику в механистических терминах; он также менее склонен описывать бездушные машины антропоморфной терминологией; следовательно, он считает вопрос, может ли машина мыслить, столь же уместным, сколь во-

прос, может ли субмарина плавать. Это общество очевидно менее тяготеет к разным техническим штучкам, отчасти потому, что не ждёт от них слишком многого, тем более спасения. Наоборот, попытка имитировать человеческий разум вызывает у них лишь комментарий: «А может, попробуете скопировать что-нибудь получше?».

Обычно мне не нужно говорить об искусственном интеллекте, поскольку это всего лишь специфическая область потенциально-го применения машин и поскольку она находится за пределами собственно информатики. Впрочем, я вынужден делать это, как только высказывается мнение, что с применением техники искусственного интеллекта машины справятся с проблемами программного обеспечения, которые нам самим не по зубам. Нашей первой реакцией на Проект Пятого По-



коления был вздох облегчения в таком духе: «Что ж, если японская промышленность пытается вложиться в искусственный интеллект, уж он-то позаботится о японской конкурентоспособности». Примерно через неделю пришло грустное понимание, что западному миру, похоже, не хватит силы духа удержаться и не примкнуть к этому поветрию.

В самом деле, он примкнул к этому поветрию, и, следовательно, опять входит в моду мысль: «не правда ли, как было бы мило, если бы наши машины были настолько умны, чтобы их можно было программировать на естественном языке?». Что же, естественные языки лучше всего подходят для своей основной цели, а именно быть неоднозначным, передавать шутки и признания в любви, но совершенно не подходят для хотя бы сколько-нибудь немного разумной точности. И если

вы не верите этому, попытайтесь прочесть современный правовой документ, и вы тут же увидите, как необходимость в точности создаёт самые неестественные языковые обороты; либо попытайтесь почитать одно из первоначальных словесных доказательств Евклида (желательно на греческом). Это исцелит вас и заставит понять, что формализмы вводятся не для того, чтобы усложнить вещи, а для того, чтобы сделать их возможными. И если вы всё ещё верите, что нам легче всего изъясняться на наших родных языках, вас следует приговорить к прочтению пяти студенческих рефератов. Проблема с «умными» машинами остаётся той же, что и со всеми «опциями» языков программирования: каждый последующий слой «дружественности к пользователю» затуманивает спецификацию и тем самым делает систему более рискован-

ной для использования.

И в заключение. Это не предмет моего доклада, однако в эти политические дни если я сам не подниму этот вопрос, он будет поднят в обсуждении: как насчёт программного обеспечения для Стратегической Оборонной Инициативы, более известной как «Звёздные войны»? Что ж, я уверен, что не смог бы его разработать, к своему удовлетворению.

Благодарю за внимание.