

Питер Наур

## Интуиция в разработке программного обеспечения

Peter Naur (1985) Intuition in Software Development // Proceedings of the International Joint Conference on Theory and Practice of Software Development, Berlin, March 1985 // Springer-Verlag, Lecture Notes in Computer Science, Vol. 186, p. 60-79.

Р. Богатырев, А. Китаев, перевод с англ.

*Интуиция характеризуется в статье как неотъемлемый компонент сознательной деятельности человека во всех ее проявлениях. Высказываются некоторые замечания относительно успешных и неудачных действий, построенных на интуиции. Приводится интуитивная основа таких общепринятых понятий, как шкала, логика, корректность, текст, обоснование и доказательство. Далее на этой основе рассматривается вопрос о том, как интуитивно строятся ключевые понятия моделей данных человеческой деятельности и разработки программного обеспечения. В заключение следует вывод о том, что при разработке программного обеспечения опыт и личностные качества программистов играют более важную роль, чем использование ими тех или иных методов.*

### Введение

Данный доклад ставит своей целью прояснить, каким образом программное обеспечение, а также отдельные понятия и приемы, используемые при его создании, воспринимаются теми людьми, которые имеют с ними дело. Точнее говоря, мы попытаемся показать, что интуиция (т.е. способность человека непосредственно постигать явления окружающего мира) — это основа, на которой должна строиться вся работа по разработке программного обеспечения.

Поводом к рассмотрению вопроса об интуиции послужила тенденция, явно наметившаяся в последнее время среди участников дискуссий по вопросам программирования. Это тенденция говорить об интуиции как о некоей человеческой слабости, которая-де приводит к большим затруднениям в разработке программ, в связи с чем программирование должно быть ограждено от какого бы то ни было влияния с ее стороны. По нашему мнению, которое будет представлено в настоящем выступлении, такой взгляд на интуицию в корне неверен. Мы утверждаем, что интуиция является неотъемлемой частью любого из видов умственной деятельности, которые и составляют в совокупности разработку программного обеспечения. В особенности это относится к таким методам, как специальная символика и различные виды аргументации. Эти методы не устраняют влияния интуиции, и их правильнее было бы считать средствами снижения вероятности возникновения ряда ошибок, свойственных интуитивному пониманию.

Нам предстоит рассмотреть довольно специфические вопросы. Можно сказать, что речь идет, главным образом, о вещах, с которыми каждый взрослый человек постоянно сталкивается в те часы, когда он бодрствует. Но эти вопросы, как правило, либо вообще ускользают от внимания исследователей, либо принимаются ими в расчет недостаточно серьезно. А когда их все же начинают анализировать более тщательно, то это, как правило, приводит к бесконечным спорам.

Мы начнем с уточнения понятия интуиции, а затем обсудим несколько других понятий, используемых в ходе разработки программного обеспечения. К ним относятся шкалы (scales), классификация, логика, корректность, синтаксис, семантика и доказательство. В заключение мы остановимся на вопросах разработки программного обеспечения и на соответствующих методах, используемых в этой деятельности.

## Понятие интуиции

Словари определяют интуицию с помощью таких выражений, как "способность к непосредственному постижению истины без логического обоснования; непосредственное постижение путем здравого смысла; непосредственное понимание". Иными словами, это понятие используется в двух различных, но связанных друг с другом смыслах, обозначая или некий процесс, или понимание, являющееся результатом такого процесса.

Зададимся теперь вопросом: почему исследователи, как правило, упускают из вида проблемы интуиции? Дело, очевидно, в том, что она буквально пронизывает все действия человека, являясь их фундаментальной составляющей, в том, что она "работает" так четко и эффективно. На протяжении всей жизни (если не считать той ее части, которая уходит на сон) мы постоянно видим и слышим, постоянно чувствуем прикосновение вещей и людей вокруг нас, узнаем их и реагируем на них. И только после некоторого размышления мы осознаем, что почти все акты действия или понимания, связанные с этими операциями, интуитивны в соответствии с приведенным выше определением. В сущности, интуиция человека охватывает его опыт, знания и память. И все эти элементы интеллектуального багажа человека интуитивны, поскольку к каждому из них мы можем обратиться буквально в любое время.

Интуиция отчасти сводится к нашей способности осмысливать впечатления, поступающие от органов чувств. Однако наше интуитивное понимание окружающей среды неотделимо от понимания языка и теоретических построений, а также от нашего сознания и мировоззрения. Обогащаясь и обновляясь, интуитивное понимание меняется на протяжении человеческой жизни. Важный аспект жизни каждого из нас состоит в приумножении своего фонда интуитивных представлений, т.е. тех видов объектов, которые мы можем узнавать интуитивно.

Именно по той причине, что интуиция пронизывает все существование человека, о ней упоминают лишь тогда, когда она проявляется неожиданно. Например, когда говорят о "женской интуиции", при этом явно имеется в виду понимание женщинами вещей, в которых они, по мнению говорящих, не должны были бы разбираться. Совсем в другом контексте трактует интуицию Мидуор (Medawar) [1], когда он рассуждает о "созидательном акте в научном исследовании", о "пришедшей в голову идее". "В естественных науках и математике, — пишет он далее, — интуиция выступает во многих формах, хотя всех их объединяют определенные общие свойства: внезапность появления, целостность воплощаемой концепции и отсутствие фазы предварительного сознательного осмысления". Аналогичные мысли мы находим у Поппера (Popper) [2]: "Мой взгляд на проблему, если он кому-то интересен, состоит в том, что не существует ни основанного на логике метода порождения новых идей, ни возможности воссоздать этот процесс с помощью чисто логических категорий. Эту мысль можно выразить иначе, сказав, что во всяком открытии есть "элемент иррациональности" или "творческой интуиции", как понимал ее Бергсон (Bergson). В том же духе высказывался и Эйнштейн (Einstein), когда он говорил о "поиске таких самых универсальных законов, ... из которых чисто дедуктивно можно вывести некую картину мира. Нет логического пути, — утверждает он, — ведущего к таким законам. К ним можно прийти лишь с помощью интуиции, основанной на чем-то вроде интеллектуальной привязанности (Einfuehlung, нем. — проникновение в сущность чего-либо; прим. переводчика) к объектам опыта".

Представление о том, что интуиция может проявляться лишь в экстраординарных обстоятельствах, стало настолько распространенным, что Куайн (Quine) [3] счел необходимым от него отмежеваться: "Дважды я с большим удивлением обнаруживал, что смысл слова "интуитивный" в моих работах искажается так, будто речь у меня идет о каких-то особых и таинственных путях познания. На деле же, говоря об интуитивной оценке, я имею в виду такое суждение, в котором термины используются в обыденном смысле, без оглядки на то, какие определения им можно было бы дать и на каких неявных предположениях они, возможно, строятся".

## Торжество и недостаточность интуиции

Пожалуй, самая важная особенность интуиции состоит в том, что нормальные взрослые люди пользуются ею с потрясающим успехом. За свою жизнь каждый человек совершает огромное число действий, постоянно вступая в контакт с окружающим миром. Каждый раз это связано с колоссальным количеством мышечных и нервных реакций, требующих невероятной слаженности.

Так продолжается день за днем, и почти всегда люди оказываются на высоте стоящей перед ними задачи.

Мы протягиваем руки, берем ими различные вещи, манипулируем ими; мы движемся — вполне успешно — среди множества других людей и объектов; с помощью наших чувств мы вступаем в контакт с более удаленными объектами окружения, узнаем предметы и людей, известных нам и неизвестных — и все это, как правило, без особых проблем.

Но наша интуиция несовершенна. Иногда она подводит нас. Мы можем уронить чашку, наткнуться на что-нибудь, обознаться или неправильно набрать телефонный номер. Такие эпизоды должны служить нам предупреждением: полагаясь исключительно на интуицию, мы рискуем потерпеть неудачу.

Есть и еще одна причина, побуждающая нас относиться к интуиции с некоторой опаской: во многих ситуациях мы слишком легко поддаемся искушению реагировать на окружающее спонтанно. Действовать на основании первой же пришедшей в голову мысли — идет ли речь о поступке или об ответе на чей-либо вопрос — кажется очень удобным, ведь любая другая линия поведения требует гораздо больших усилий.

Дело осложняется еще и тем, что в типичных жизненных ситуациях реакции человека невозможно классифицировать как успешные или неудачные. Иными словами, в ряде случаев отсутствует сама основа для оценки интуитивных реакций. Это справедливо в первую очередь для реакций, связанных с пониманием языка. Ведь наше понимание почти всегда абсолютно интуитивно, к тому же оно развивается на протяжении всей нашей жизни, что объясняется, помимо прочего, развитием самого языка.

### **Насколько же несовершенна интуиция?**

Одно из главных положений настоящего доклада состоит в том, что во всех своих действиях люди всегда должны полностью полагаться на интуицию; с другой стороны, при этом нет возможности абсолютно исключить появление ошибок, так что в лучшем случае можно лишь сгладить их неблагоприятные последствия. В этом разделе мы как раз и рассмотрим некоторые основные методы, позволяющие добиться подобного сокращения ущерба.

Бывает, что ошибки интуитивного восприятия непоправимы: если мы уронили чашку и она разбилась, с этим ничего нельзя поделать. В других случаях (если мы, скажем, неверно набрали телефонный номер) положение можно исправить, проведя операцию повторно. Уже этих примеров достаточно, чтобы сделать два вывода, важных с точки зрения успешного применения интуиции: во-первых, очень важно отличать более серьезные ошибки от менее серьезных, и, во-вторых, нужно иметь в виду, что в ряде ситуаций допущенную ошибку можно исправить путем повторения определенной операции вплоть до получения желаемого результата. Возможны и другие варианты: повторение операции с последующей проверкой результатов на тождественность, тактика осторожных и медленных действий, привлечение других людей для проверки полученных результатов.

Следует отметить, что осмысленное применение этих приемов требует интуитивного соотнесения действия с более широким контекстом, в котором оно совершается. Они, можно сказать, демонстрируют, как интуитивное понимание одного аспекта проблемы может подкрепляться интуитивным пониманием другого аспекта, в результате чего вся работа приобретает осознанный характер. Таким образом, вышеизложенные приемы вполне вписываются в рамки научного подхода, который Куайн (Quine) [3] определяет такими словами: "наука есть осознавший себя здравый смысл". Чтобы успешно применять эти тесно связанные между собой приемы, нужно быть самокритичным, придерживаться этих недогматичного взгляда на вещи и всегда быть готовым признать, что любое интуитивное понимание имеет лишь ограниченную ценность.

### **Интуиция, шкалы, логика и корректность**

До сих пор мы рассматривали проблемы интуиции, намеренно оперируя такими расплывчатыми понятиями, как торжество и недостаточность (успех и неудача), не пытаясь дать им более точное определение. При этом мы исходили из того, что человек, опираясь на интуитивное понимание, вполне способен строить свои отношения с окружающим миром без какой-либо базы в виде критериев или шкал значений.

Мы полагаем, далее, что практическое использование хорошо узнаваемых категорий, чисел и мер основывается исключительно на нашем интуитивном понимании. Это означает, в частности, что мы можем допускать применение определенных категорий, чисел и мер, даже не имея ясного представления о возможных ограничениях такого применения, как в плане точности определений, так и в отношении области применения. Это особенно справедливо для таких случаев, когда объекты разделяются на две взаимоисключающие категории: правильное и неправильное, "то, что нужно" и "не то, что нужно". Возьмем такой пример: отправляясь в путешествие, мы можем сказать, что сели на нужный нам поезд (the right train) и приехали именно в тот город, куда намеревались (the right city). Поезда и города обычно, хотя и не всегда, достаточно дискретны. Однако мы, как правило, не можем сказать, оказалась ли "такой" или "не такой, как надо" погода или отношение к нам со стороны местного населения. Более того, мы даже не можем обозначить показатели погоды и доброжелательности населения на какой-либо шкале значений. А ведь при всем при этом наши представления о погоде и доброжелательности вполне понятны на интуитивном уровне.

Эти соображения показывают, что любые описания окружающего мира с помощью строгих категорий и всевозможных шкал значений ничуть не содержательнее, нежели наши интуитивные представления о нем.

Из сказанного следует, далее, что когда люди допускают нарушения, по определению выражающиеся в несоблюдении неких жестких ограничений, эти нарушения и ошибки в определенном смысле вызваны установлением упомянутых ограничений — в той же степени, что и конкретным ошибочным действием. Если, скажем, не существует норм, регламентирующих поведение людей на проезжей части улиц, о нарушениях правил дорожного движения говорить просто не приходится.

Когда дело доходит до применения логики к явлениям окружающего мира, мы обнаруживаем, что и здесь роль интуиции исключительно велика. Чтобы проиллюстрировать эту мысль, давайте рассмотрим расхожее среди логиков утверждение о том, что наше знание состоит из предикатов. Возьмем для примера предложение: "Париж — столица Франции". С точки зрения логика оно по сути является предикатом, то есть утверждением, которое может быть либо истинным, либо ложным. Но всем известно (это часть нашего интуитивного знания), что смысл данного предложения гораздо шире. Если мы зададимся вопросом, почему можно согласиться с тем, что Париж — столица Франции, то придем к выводу, что дело здесь не в каком-то строгом логическом доказательстве. В пользу справедливости этого утверждения мы, скорее всего, приведем множество свидетельств, ни одно из которых не претендует ни на необходимость, ни на достаточность. Мы скажем, к примеру, что в Париже находится резиденция президента Франции, что там расположены министерства французского правительства и работает парламент страны, что в каждой книге о Франции Париж упоминается как ее столица и т.д. и т.п.

В языке, которым мы пользуемся в повседневной жизни, понятия, близкие логическим категориям истинности и ложности, встречаются редко. Знание того факта, что Париж — столица Франции отнюдь не равнозначно признанию истинности предиката в данном утверждении. Наше интуитивное знание, которое проникает в область научного знания таким образом, что между ними нет четко выраженной границы, представляет собой совокупность связанных между собой бесчисленных элементов понимания и ноу-хау. Именно они помогают нам адаптироваться к постоянно меняющейся ситуации. Поэтому соглашаясь с тем, что Париж — столица Франции, мы опираемся на свои знания о том, как люди группируются в города и государства, а быть может, — на теорию государственного устройства и наши представления о важной роли столиц. Эти знания позволяют нам объяснить, почему мы признаем справедливость такого утверждения и, возможно, указать на обстоятельства, при которых мы не смогли бы этого сделать. Вопросы об истинности и ложности здесь вообще не возникает. Такой вопрос уместен разве что в особых ситуациях, где задача состоит в том, чтобы выяснить, лжет или говорит правду тот или иной человек.

Итак, логика связана с интуицией, и эта связь затрагивает не только прикладную логику, он и саму внутреннюю структуру этой науки. В сущности, теоретические категории (даже когда речь

идет о самых авторитетных и стройных системах) могут быть поняты человеком только интуитивно.

Уайтхед (A.N.Whitehead) [4, с.266] указывает, что у разных современных авторов, весьма искусственных в предмете, такие фундаментальные понятия, составляющие формальную логику, как высказывание (proposition), наделяются множеством разных значений. Заметим, к слову, что и само утверждение Уайтхеда основано на чистой интуиции: ведь прийти к выводу о нетождественности смысла одного и того же понятия в разных контекстах можно только интуитивно.

О такой категории, как корректность, можно сказать то же, что было сказано об истине. Мы не спрашиваем, работает ли корректно автомобиль; вместо этого мы употребляем выражение "машина в рабочем состоянии". Такая формулировка не предполагает соблюдения неких жестких критериев и вполне допустима в том случае, когда нам известно о мелких неисправностях автомобиля. Она означает лишь, что в достаточно широком диапазоне внешних условий наша машина способна выполнять те функции, которые и должен выполнять автомобиль. Если мы попытаемся дать более точную формулировку, то убедимся, что это невозможно. Для того чтобы установить четкие критерии, необходимо иметь точные параметры, тогда как работа машины в значительной степени определяется характеристиками, в такие параметры не вписывающимися. Ну а если для создания критериев мы все-таки примем произвольные параметры, у нас не будет гарантии того, что автомобиль, безупречный по нашим меркам, не выйдет из строя при нагрузках, которые водитель сочтет вполне нормальными.

Это наблюдение имеет прямое отношение к проблеме корректности программ. Ситуацию можно описать следующим образом. Ясно, что пользователь программы хочет, чтобы эта программа работала, то есть давала нужные результаты. Однако это его желание не является (да и не может быть) основой для строгого, раз и навсегда данного описания требований пользователя. В самом деле, хотя программа будет несомненно давать такие результаты, которые пользователь без колебаний сочтет корректными или некорректными, чаще всего при это будут отмечаться целые классы других результатов или реакций со стороны программы — таких, к которым пользователь просто не готов и которые он не в состоянии оценить. На вопросы, касающиеся подобных случаев, значительная часть пользователей может ответить — и вполне обоснованно — что многие из них никак не связаны с нормальным выполнением программы и потому вообще не должны иметь место.

В такой ситуации некоторые специалисты в области компьютерных наук утверждают: если подходящий критерий корректности отсутствует, его должен ввести программист. Для достижения этой цели ему следует описать действие разрабатываемой программы с помощью набора четко определенных спецификаций. И если действие программы соответствует данным спецификациям, она считается "корректной". Что ж, здесь все правильно. Но все же такой подход не отменяет того положения, что и совместимость текстов, и другие внешние обстоятельства могут восприниматься только интуитивно. Так что вопрос о том, насколько тексты спецификаций совместимы с важными для пользователя обстоятельствами окружения, все равно можно разрешить только интуитивно.

### **Интуиция и текст**

Значительную роль в разработке программного обеспечения играет работа программиста с текстами (в понятие текста здесь включаются любые данные в пригодной для чтения форме: проза на обычном языке, программы и формулы всех видов).

Поскольку текст позволяет в относительно стабильной форме сохранять информацию, которую не всегда способна удержать наша память, он может служить средством компенсации недостатков человеческой интуиции, и в этом состоит одна из его функций в контексте обсуждаемой нами проблемы. Рассмотрение этой функции заставляет нас обратиться к тому, каким образом текст воспринимается интуитивно и как он влияет на знания человека.

В последнее время участники дискуссий по проблемам текста и языка все чаще принимают как нечто само собой разумеющееся мысль о том, что при работе с текстами следует различать две вещи — синтаксис и семантику. Сформировалось даже твердое убеждение: чтение текста должно

сопровождаться анализом его синтаксического и семантического аспектов. В кругах, связанных с вычислительной техникой, это убеждение подкрепляется еще и тем обстоятельством, что как в проектировании, так и в самой структуре компиляторов языков программирования выделение синтаксического и семантического играет полезную роль. В такой ситуации может показаться очевидным, что и человек, воспринимая тексты, должен каким-то образом проводить такое разграничение между синтаксисом и семантикой.

Надо отметить, что предположение о взаимной обособленности, якобы присущей таким категориям, как синтаксис и семантика, строится на нескольких сомнительных посылах. Первая — будто вообще возможно проводить синтаксический анализ текста независимо от семантического анализа. И вторая — будто интуитивное восприятие текста (а именно так читает тексты человек) основано на таком обособлении.

Относительно первой посылки стоит сказать, что такие концепции, как синтаксис, семантика и некоторые другие были введены в научный оборот главным образом для того, чтобы облегчить описание языковых явлений. Но вот вопрос о том, какие концепции следует использовать для описания каждого конкретного языка, никоим образом нельзя считать решенным раз и навсегда. Ведь выбор концепций определяется простотой описания и удобством применения к конкретному языку и, конечно же, отражает точку зрения лингвиста.

Например, Есперсен (Jespersen) в своей классической работе по грамматике английского языка [5] пишет, что "грамматика обычно подразделяется на две части: морфологию, изучающую все формы языка (флексии), и синтаксис, изучающий структуру предложений и правила использования этих форм". Но далее автор разъясняет, почему в своей книге он отступает от этой классификации.

Если взглянуть на вещи в более широком контексте и представить себе, как происходит первое знакомство человека с языком, придется констатировать, что проблема синтаксиса и семантики здесь вообще не стоит. В лучшем случае эти концепции можно использовать для разъяснения некоторых особенностей языка лицу, уже овладевшему им на интуитивном уровне. Подобным же образом, применение этих концепций при анализе текста имеет смысл лишь для того читателя, который уже понял содержание текста. Это соображение применительно к математике было высказано Кляйном (Kline) [6], который пишет: "К изложению любого нового предмета педагог всегда должен подходить интуитивно. Строго логическое обоснование — это всего лишь искусственная реконструкция того, что разум напрямую усваивает из рисунков, физических опытов, особых случаев и банальных проб и ошибок. Теория исчисления играет ничуть не большую роль в понимании самого предмета, чем химическая теория окисления — в овладении техникой вождения автомобиля". Цельность, комплексность интуитивного усвоения текста человеком особенно ярко проявляется в тех случаях, когда мы легко схватываем текст, который при строгом соблюдении правил чтения уяснить просто невозможно. В то время, когда писались эти строки, в газетах появилась такая реклама фильма, вышедшего на экраны Копенгагена: En Jusser i New York.

Пожалуй, любой нормальный датчанин немедленно поймет, что здесь написано по-датски "Русский в Нью-Йорке". Но всякая попытка проанализировать этот текст "по всем правилам" неизбежно завершится неудачей на первом же этапе — на уровне распознавания отдельных букв, ибо уже второе слово начинается с буквы, которой вообще нет в датском алфавите. Истинное, реальное понимание реализуется через взаимодействие процессов, которые при анализе по правилам оказываются на совершенно разных уровнях, не говоря уже о том, что здесь задействована способность выявления сходства и аналогий, лежащая вне сферы всяких правил.

Вывод о том, что интуитивное понимание человеком текста представляет собой цельный акт, подтверждается предпринятым Ледгардом (Ledgard) и др. исследованием интерфейсов интерактивного текстового редактора [7]. Как подчеркивают авторы сообщения, "участники эксперимента не проводили различия между синтаксисом и семантикой. Им даже не приходило в голову, что способность выполнять операции по редактированию и что функция и появление текущих команд — это разные вещи. Идея о том, что текущие команды как раз и представляют собой редактор, была для них совершенно очевидной, и когда после эксперимента им сообщили, что два редактора функционально идентичны, многие из них были искренне удивлены".

Сейчас наша главная задача состоит в том, чтобы показать, сколь важную роль играет интуиция в процессе восприятия текста человеком, а этого нельзя понять, если рассуждать в категориях

"чтения по правилам", таким, как правила синтаксиса. Эта мысль станет яснее, если мы осознаем, что, читая текст, человек совершает некое неделимое действие. При этом он принимает множество решений: действительно ли то, что находится у него перед глазами, является текстом; имеет ли этот текст отношение к цели его работы, скажем, к разработке программного продукта; из каких элементов — утверждений, формул — состоит этот текст и т.д. Так вот, эти решения тоже не могут быть разделены как посылки в цепочке доказательств; они просто вытекают из интуитивного понимания. Эту мысль проще всего пояснить, показав, что если бы данные решения принимались по отдельности, строго по правилам, нам потребовалась бы бесконечная цепочка правил — сначала тех, что регламентируют принятие решений, затем тех, что определяют порядок применения соответствующих правил и т.д. Вернемся к первому решению: является ли текстом то, что находится у нас перед глазами? Если бы существовали отдельные правила для принятия такого решения, немедленно возник бы вопрос: а как решить, что эти конкретные правила применимы к данному случаю? Так появляется потребность в новых правилах, а вслед за ней — потребность в правилах, определяющих, какие из новых правил применяются в каких случаях — и т.д. до бесконечности. Поскольку это абсурдно, от первоначальной посылки ("вопрос о том, является ли данный объект текстом, можно решать с помощью некоторых правил") приходится отказаться.

На основе изложенного можно сделать следующий вывод: в процессе восприятия текста человеком ни один из аспектов этого текста не играет решающей роли, не является более важным, чем другие аспекты. Так, в частности, повсеместное применение синтаксического сахара (*designation syntactic sugar*) для указания на определенные аспекты символики предполагает противопоставление их другим аспектам, якобы более важным, а это часто бывает неверно. Следовательно, при оценке достоинств языка или символики, с помощью которых поддерживается разработка программного обеспечения, следует принимать в расчет весь диапазон их применения в конкретной ситуации и конкретным программистом. Иными словами, главный критерий: "работает" ли избранная нами символика — нельзя заменять любым другим критерием, основанным на значениях шкалы формальных характеристик.

### **Обоснование, доказательство и интуиция**

Связь между обоснованием, доказательством и интуицией имеет решающее значение в контексте обсуждаемой нами проблемы. Словари определяют обоснование (*reasoning*) как процедуру, позволяющую на основе некоторых посылок сформулировать или прийти к определенным выводам посредством связного мыслительного процесса, не проявляющегося внешне или выраженного каким-либо образом. (Доказательство, соответственно, — это итог успешного обоснования.) В рамках нашего изложения важно отметить, что и обоснование, и доказательство не только не исключают интуицию, но только не противопоставляются ей, но и вообще невозможны без участия интуиции. Человек может предложить обоснование, а другой человек — понять содержащееся в нем доказательство лишь в том случае, если оба они имеют интуитивное представление о каждой посылке, об их взаимосвязях и о том, каким образом из этих посылок следует данный вывод. Разумеется, нужно еще и интуитивное понимание того, что же такое доказательство. Поэтому можно сказать, что доказательство — это выражение интуитивного понимания ряда обстоятельств, взятых в их взаимосвязи и упорядоченных таким образом, чтобы продемонстрировать, как из обоснования следует вывод.

Давайте возьмем пример, связанный с такими проблемами реальной жизни, которые представляют интерес и для создателей прикладных программ. Допустим, нам требуется доказать, что сейчас в корзине находится четыре яблока. Итак, доказательство: минуту назад я видел, что корзина была пуста; потом я заметил, как сначала Съюзен, а за ней Барбара положили в нее по два яблока; больше ничего не происходило; а дважды два — это четыре.

Наш пример весьма прост, но в нем содержатся все элементы, составляющие доказательство факта реальной жизни, и он показывает, что в основе всего доказательства лежит интуиция. Ведь чисто интуитивными были и решение о том, что корзина пуста, и фиксация наличия яблок в руках у обеих девушек, и соотнесение этого факта с натуральными числами, и вывод о том, что в данной ситуации нужно прибегнуть к операции умножения. При более тщательном анализе может встать вопрос, при каких условиях доказательство сохраняет свою силу. Отвечая на него, можно сказать, что корзина должна быть обычной, без дыры в днище и что доказательство справедливо лишь в течение ограниченного времени — до тех пор, пока одно из яблок не сгниет. Кстати, какой-нибудь особенно осторожный респондент, возможно, захочет добавить, что в основе всего

доказательства лежит посылка, согласно которой вселенная представляет собой своего рода континуум, но такое предположение, пожалуй, уже выходит за рамки ответа на поставленный вопрос.

Наш пример показывает также, что для доказательства требуется логическая модель того или иного аспекта окружающего мира (в данном случае речь идет о возможности применения целых чисел при подсчете таких элементов этого мира, как яблоки). Подобную модель с вытекающим из нее соответствием между определенными объектами (в нашем случае — яблоками) и логическими конструкциями (числами) человек может усвоить только интуитивно. Часть этого интуитивного знания сводится к тому, что данная модель применима к яблокам, по крайней мере, при определенных обстоятельствах. А это важно потому, что такая модель не всегда будет работать. Ведь если, скажем, мы поместим в пустую чашку две капли воды, а потом добавим еще две капли, совсем необязательно, что в итоге на дне чашки окажется четыре капли воды.

Доказательства, основанные исключительно на свойствах абстрактных (например, математических) понятий, имеют особый характер. Дело в том, что некоторые абстрактные понятия по определению пребывают в вечной неизменности и потому на их основе можно получать результаты, имеющие силу на протяжении неограниченного времени. Это свойство, конечно, уникально для нашего мира, где все остальное постоянно изменяется. И все же такое свойство абстракций не отменяет того факта, что доказательства строятся на основе интуиции. Ведь для построения доказательства нужно уяснить интуитивно как входящие в него абстрактные понятия, так и их взаимосвязи.

Но если обоснование полностью основывается на интуиции, то интуитивное понимание лишь в редких случаях опирается на обоснования и доказательства. На протяжении всей нашей жизни наши интуитивные знания постоянно развиваются, поступающие от наших органов чувств впечатления объединяются с теми знаниями, которыми мы уже располагали, образуя в итоге элементы нового интуитивного знания. Процессы объединения нового понимания, новых наблюдений с уже имеющимся знанием проходят по преимуществу — и мы это чувствуем — без особых усилий, здесь нет нужды опираться на законы, на принципы или строить алгоритмы. Однако при более внимательном рассмотрении мы неизбежно приходим к выводу, что не в состоянии обнаружить каких-то четких различий между этими процессами и обоснованием. Просто об обосновании как таковом мы говорим лишь в тех случаях, когда число входящих в него обстоятельств и связей между ними достаточно велико, а вывод следует лишь после довольно тонкой их комбинации, вследствие чего формулирование доказательства представляется имеющим смысл.

Когда мы знакомимся с каким-либо доказательством, это, возможно, и помогает нам интуитивно понять вывод, но подобное знакомство не является ни необходимым, ни достаточным условием такого понимания. Ведь можно принять все этапы доказательства и не суметь при этом интуитивно уяснить для себя вытекающий из него вывод. И наоборот, мы можем принять справедливость какого-либо утверждения бездоказательно. Иногда случается так, что человек помнит вывод, но не помнит соответствующего доказательства. Вполне реальна, например, такая ситуация. Утверждение "любое квадратное уравнение имеет не более двух различных корней" стало частью интуитивного знания человека, изучавшего когда-то элементарную математику, — но он забыл, как оно доказывается. С другой стороны, для того чтобы понять это утверждение, мы должны обязательно знать интуитивно такие элементы, как квадратные уравнения, корни, различие между корнями, не говоря уже о знании того, что представляют собой числа.

Интуитивные знания, в формировании которых доказательства сыграли определенную роль, по сути своей не отличаются от знаний, полученных вне связи с логическими доказательствами. Если доказательство и может привнести в наши знания нечто новое, так это интуитивное понимание взаимосвязей между отдельными элементами интуитивного знания. Понимание того, каким образом посылки объединяются в доказательстве, важно уже само по себе. Это следует из того факта, что человек может хорошо понимать все посылки, но при этом быть не в состоянии сделать нужный вывод. А это случается сплошь и рядом, и даже в таких сферах деятельности, где выдумка и изобретательность играют очень важную роль.

Подведем итог: вне зависимости от интуиции формирование знаний с помощью обоснований и доказательств невозможно, ибо фактически эти понятия представляют собой особый метод упорядочения интуитивных представлений.

## Модели данных человеческой деятельности

Программное обеспечение разрабатывается для того, чтобы облегчить выполнение людьми тех или иных видов деятельности. И хотя способы поддержки могут принимать множество различных форм и иметь отношение ко многим сторонам жизни человека, общий основополагающий принцип программных решений состоит в использовании модели человеческой деятельности, построенной в форме данных и процессов данных. Чтобы показать связь между понятиями "данные", "процессы данных", "спецификации", "представления данных", "формализация" и "доказательство", мы коротко рассмотрим один из видов человеческой деятельности, а также то, как эта деятельность поддерживается с помощью модели. Хотя вид деятельности, избранный нами, достаточно прост, на его примере хорошо видны типичные проблемы, возникающие при проектировании моделей данных.

Итак, для примера мы возьмем такой вид деятельности, как пополнение запасов необходимых товаров в обычной семье. Один из способов решения этой проблемы состоит в том, что пополнение запасов осуществляется с регулярными интервалами, т.е. все необходимое закупается в ходе очередного закупочного цикла. Это означает, что каждый раз, отправляясь по магазинам, отвечающий за закупки член семьи должен решить две задачи. Во-первых, выявить потребности на основе оставшихся запасов и предполагаемого объема потребления вплоть до следующего закупочного цикла, и, во-вторых, приобрести и доставить домой необходимые товары. При реализации операции закупки в соответствии с такими общими принципами могут оказаться полезными спецификации и формализованные описания. Так, объем закупок в ходе каждой поездки по магазинам можно специфицировать с помощью плана запасов (stock plan), то есть списка, в котором на основании таких показателей, как усредненный объем потребления в семье, срок хранения того или иного продукта и объем кладовой указывается минимальное и максимальное количество по каждому наименованию товаров. На первой стадии операции закупщик сопоставляет эту спецификацию с данными по остаткам каждого вида товаров и получает в итоге список необходимых товаров, то есть формальное описание того, какие виды товаров и в каком количестве должны быть приобретены. На второй стадии с помощью этого списка производится отбор товаров с полок магазинов.

Анализ принципов и действий, связанных с пополнением домашних запасов, показывает, что к помощи интуиции приходится прибегать на всех стадиях этого процесса - как при его планировании, так и при выполнении.

Кое-кто, пожалуй, не согласится с этим и заявит, что всю процедуру можно систематически вывести из единого универсального требования: не допускать полного исчерпания запасов того или иного продукта. Но при этом нужно заметить, что в самом этом утверждении определение требования основывается на интуитивном понимании ситуации в целом — и в той ее части, которая касается семьи, и применительно к таким категориям, как запасы и разные виды товаров. Ведь разрабатывая принципы универсальной процедуры для операции по закупке, мы опираемся на интуитивное понимание процесса потребления товаров в семье и применимости к нему формального понятия усредненного потребления. Это интуитивное понимание особенно важно при установлении связи между самой процедурой закупки и универсальным требованием, на котором она должна строиться. Далее, такое понимание имеет решающее значение при определении области применения процедуры и при расчете для нее параметров "запаса прочности", а также мер безопасности, позволяющих использовать эту процедуру в некоторых нестандартных ситуациях. В то же время должно быть очевидно, что никакая процедура не может обеспечить выполнение основного требования при всех мыслимых обстоятельствах. Если, к примеру, семейные запасы какого-то товара исчезают после ограбления или сгорают при пожаре, это значит, что в данный момент главное требование уже не удовлетворяется.

При осуществлении процедуры пополнения семейных запасов интуиция "выходит на сцену" всякий раз, когда человек каким бы то ни было образом вступает в контакт с тем или иным товаром, прежде всего — когда он идентифицирует этот товар или, возможно, анализирует его более тщательно в категориях количества и качества. Затем интуитивное понимание, возникшее в ходе этого прямого контакта с товаром, возможно, будет сопоставлено с формальным описанием, содержащимся в плане запасов и в списке необходимых товаров.

Если же при разработке процедуры пополнения семейных запасов приходится использовать формальные доказательства, это опять-таки связано с интуитивным пониманием в нескольких отношениях. Такая процедура может, к примеру, включать алгоритм определения минимального и максимального количества товаров, который будет использоваться при составлении плана

запасов. А может быть, при разработке процедуры потребуется доказать, что данный алгоритм гарантирует удовлетворение исходного универсального требования. Этот этап разработки может увенчаться успехом лишь при условии интуитивного понимания того, что критерии такого доказательства действительно обеспечивают правильность вывода, вытекающего из доказательства. Более того, в числе посылок такого доказательства обязательно будут фигурировать некоторые условия, ограничивающие параметры моделируемой ситуации в известных пределах. Так вот, вопрос о том, удовлетворяются ли, и могут ли в принципе быть удовлетворены эти ограничительные условия (а если да, то в какой степени) способен решить только проектировщик на основе своего интуитивного понимания.

### Разработка программного обеспечения

В ходе нашего обсуждения проблемы важности интуиции в разработке программного обеспечения решающее значение имеет правильный взгляд на процесс разработки, понимаемый как взаимодействие интуитивных знаний программиста и его реального окружения, которое включает используемые или создаваемые им тексты. В последующем изложении термин "знания" мы будем применять для обозначения интуитивных знаний человека. Однако при этом возникает одно обстоятельство, порождающее множество трудностей, а то и неразбериху. Дело в том, что нам придется говорить о частях интуитивного знания, а ведь в то же время мы не можем не признать, что знания человека в любой данный момент представляют собой неделимое целое. Так вот, признание нами упомянутой истины найдет свое отражение в том, что идентификаторы (K1, K2 и т.д.), которые мы введем в последующем тексте для обозначения знаний программиста, будут указывать не на их отдельные фрагменты, а на знания, взятые в их совокупности и в динамике их приращения в процессе разработки программного обеспечения.

Реальное окружение (the real world), открытое взору программиста и других людей, будет обозначено символом W. Как интуитивно признают и программист, и другие люди, природа отдельных частей этого окружения имеет собой характер, свойственный написанным на определенном языке текстам, или символическому. Такие элементы окружения будут обозначены символами WT1, WT2 и т.д.

Теперь работа по разработке программного обеспечения в максимально упрощенном варианте может быть описана с помощью следующих элементов:

- W. Реальное окружение, доступное для наблюдения несколькими людьми.
- WT1. Текст, описывающий язык программирования.
- WT2. Текст, описывающий проблему, которую должна разрешить разрабатываемая программа.
- WT3. Текст программы и ее документация.
- K1. Знание обстановки, обычного языка и применяемого языка программирования.
- K2. K1 в совокупности со знанием разрешаемой проблемы.
- K3. K2 в совокупности со знанием программного решения или с теоретическим представлением о нем.

Работа или процесс разработки программного обеспечения обычно протекают следующим образом. Первоначально программист обладает элементом K1, при том, что имеются в наличии также элементы W и WT1. При этом следует предположить, что элементы K1 и WT1 состоят между собой в некотором тесном отношении интуитивной совместимости — в том смысле, что человеку, обладающему элементом K1 и познакомившемуся с элементом WT1, последний покажется знакомым и понятным, по крайней мере, в большей своей части.

С другой стороны, было бы заблуждением утверждать, что K1 содержит лишь нечто вроде копии WT1. Возможно, конечно, что, обретая K1, человек использовал WT1 как средство получить представление о данном языке программирования, но ведь формирование K1 в значительной степени основывалось также на полученных ранее знаниях о программировании и вообще об окружении и языке, а также на доступности W.

Работу по созданию программного обеспечения можно инициировать, предоставив в распоряжение программиста WT2. Этот элемент будет одним из важнейших источников формирования у программиста элемента K2, который, впрочем, неизбежно будет в значительной

степени основываться и на K1. В самом деле, ведь даже понять поставленную перед ним проблему по программированию человек может лишь в том случае, если он обладает предварительными знаниями о программировании и о языке программирования.

Столь же важная задача программиста при формировании K2 состоит в том, чтобы соотнести содержание WT2 со своим знанием окружения, что составляет часть элемента K1; возможно, что кроме этого ему придется обратиться и к W. Такие комбинации элементов реального окружения и знания мыслимы лишь как проявления интуитивного знания. В самом деле, чтобы соотнести текст (например, WT2) с окружением W, надо с одной стороны понять и интерпретировать этот текст, а с другой — отобрать определенные элементы окружения и присвоить им некоторое значение. Вся эта работа имеет какой-то смысл лишь в том случае, если она воспринимается как предпринятые программистом продуманные, целенаправленные действия, основанные в каждой своей стадии на интуитивном знании.

Фактическая разработка реального программного обеспечения, если она ведется строго в виде последовательности рабочих фаз, состоит в том, что программист формулирует теорию решения и получает таким образом элемент K3, а затем пишет W3 — текст программы и ее документацию.

Если рассматривать проблему под этим углом зрения, то единственным источником текста программы W3 является полностью сформированное знание K3 как выражение определенного аспекта этого интуитивного знания.

Однако на деле проектирование программного обеспечения очень редко происходит в соответствии с описанными в предыдущих параграфах фазами, если вообще оно так когда-нибудь ведется. Поэтому, надо полагать, строгое разграничение между K2 и K3 возможно лишь в редких случаях. Гораздо более вероятно, что процесс проектирования примет вид постоянного взаимодействия, участниками которого являются с одной стороны - программист, все более глубоко уясняющий и саму проблему, и пути ее решения, а с другой - элементы реальной обстановки W, WT2 и WT3. По мере продвижения работы над документацией и программой интуитивные знания программиста, как правило, углубляются, а формирование элемента K2 происходит уже в свете нового взгляда на доступные элементы W и WT2. Случается, что программисту приходится также обращаться к WT1, описанию языка программирования, хотя эффективно решать свою задачу он может лишь в том случае, если его представления о языке программирования уже даны в основном в элементе K1.

В конце концов неважно, протекает ли процесс разработки программного обеспечения в виде последовательности отличных друг от друга фаз или он представляет собой одну более сложную фазу — тот факт, что интуиция программиста играет при этом решающую роль, не подлежит никакому сомнению. Ибо на любом этапе задача состоит в том, чтобы находить нужные сочетания таких элементов окружения, которые не имеют внутреннего единства вне интуитивного понимания программиста, вне осознания им цели деятельности и значения этих элементов в контексте ее достижения.

## **Интуиция и методы разработки программного обеспечения**

В контексте данного раздела под методом разработки программного обеспечения мы будем понимать совокупность правил, направляющих действия программиста по решению стоящей перед ним задачи и выходящих за рамки тех внутренне присущих процессу программирования правил, которые описаны нами в предыдущем разделе. Правила, составляющие некоторый метод, обычно связаны отношениями взаимной зависимости и поддержки. Наиболее важные проблемы, касающиеся метода, можно сгруппировать в три крупных категории, как это показано ниже.

**1. Действия (activity).** Правила, определяющие, какие именно операции необходимо совершить в дополнение к минимальному комплексу мероприятий по разработке программного обеспечения. Для этой категории характерны такие действия: описание переменных, установление инвариантных свойств переменных выполнения программы; описание программной логики в формах, отличных от самой программы, например, в виде спецификаций отношений между вводом и выводом; доказательства непротиворечивости или совместимости определенных выражений; некоторые операции проверки; сквозной контроль.

**2. Формы выражения (forms of expression).** Различные символика или языки, которые должны применяться с различными целями при разработке программного обеспечения.

**3. Порядок действий (ordering of activities).** Установление определенной очередности выполнения операций в ходе разработки программного обеспечения, например, нисходящее проектирование или пошаговое уточнение.

Одна из важных проблем, возникающих при рассмотрении того или иного метода разработки, состоит в том, насколько этот метод эффективен; иными словами, способствует ли он, а если да, то в какой степени, улучшению качества разработки, в ходе которой этот метод применяется.

В принципе, эффективность методов желательно устанавливать эмпирически, исследуя реальный процесс разработки программного обеспечения. Однако, применяя такой подход на практике, мы сталкиваемся с разного рода сложностями. Как, к примеру, определить, какие действия по разработке программного обеспечения можно взять за основу для сравнения? Каким образом следует оценивать и соизмерять разного рода улучшения в качестве разработки? Как можно гарантировать, что такие реально обнаруженные улучшения действительно являются результатом применения данного метода?

Вообще говоря, эмпирический анализ метода, применяющегося для исправления результатов некачественно проведенной работы, всегда осложняется проблемой, которую описавший ее Бернард Шоу (Bernard Shaw) [8] назвал парадоксом исправления халатности (Surprises of Attention and Neglect): когда сделанная кое-как работа приводит к тяжелым последствиям, любое исправление, даже если оно базируется на совершенно неверных посылах, скорее всего, поможет улучшить ситуацию. Понятно, что в таких обстоятельствах улучшения, полученные в результате применения того или иного метода, отнюдь не могут служить достаточным доказательством эффективности самого метода.

Мы предлагаем здесь в качестве альтернативы и дополнения к эмпирической оценке методов разработки программного обеспечения, оценивать эти методы на основе их отношения к интуиции. Для начала можно отметить, что, как и в случае со всеми другими действиями по разработке программного обеспечения, применение правил в рамках того или иного метода непосредственно связано с интуитивным решением программиста, который определяет, когда и как применять каждое правило, входящее в арсенал используемого в данном случае метода.

Второе важное обстоятельство состоит в том, что все эти различные правила, составляющие тот или иной метод, должны быть соотнесены с недостатками, характерными для интуитивного знания и поведения программиста. А недостатки эти бывают двух видов:

- (1) программист по забывчивости не совершает действий, которые должны быть совершены в соответствии с его интуитивным пониманием;
- (2) программист допускает ошибки в действиях, совершаемых интуитивно.

Говоря о недостатках программиста, вызванных забывчивостью, весьма уместно вспомнить о первой категории проблем, касающихся метода (мы обозначили ее как действия). Это может оказаться полезным. В самом деле, обычный контрольный список действий, которые, как показывает опыт, часто применяются при разработке программного обеспечения, может стать эффективным средством борьбы с такими "пропусками по забывчивости". Составление подобного списка (и, разумеется, обязательное использование его в работе) можно считать частью метода в данной сфере деятельности.

Вторая группа проблем, касающихся метода (формы выражения), не имеет прямого отношения к забывчивости программиста. Однако третья группа (порядок действий) все же имеет к этой проблеме некоторое отношение.

Ведь даже такой простой инструмент, как контрольный список, нужно использовать не на самых последних стадиях разработки, если мы и в самом деле хотим получить от него хоть какую-то пользу.

Теперь об ошибках, которые программист допускает в действиях, совершаемых интуитивно. Здесь очень важно то, что любой метод предусматривает, помимо всего прочего, проведение программистом проверок. Такие проверки могут сводиться к простому повторению какого-то

этапа работы с тем, чтобы удостовериться в идентичности результатов. Однако контрольный аспект неявно присутствует и в некоторых других видах действий, включая описание программной логики в формах, отличных от самой программы. Сравнивая прямые и неявные проверки, мы обнаружим, что и те, и другие имеют свои достоинства и недостатки.

Преимущество прямых проверок, основанных на повторении действий, состоит в том, что не возникает сомнений относительно объекта проверки. С другой стороны, поскольку такие проверки осуществляются одним и тем же лицом, риск повторения совершенной однажды интуитивной ошибки довольно велик; в этом-то и состоит их недостаток. Его можно устранить, поручив проверку другому программисту. При неявных проверках интуитивные ошибки исключаются, но возникают новые проблемы: насколько такие проверки эффективны при верификации решений? Обычно в ходе проверки удается проконтролировать лишь некоторые аспекты решения, поэтому такие детали, как правильность написания имен переменных почти никогда не проверяются в полном объеме.

Применение определенного метода налагает на программиста обязательство использовать соответствующие формы выражения. Это обстоятельство, с одной стороны, и интуитивные ошибки, с другой, связаны между собой довольно сложными отношениями. Часто приходится слышать, что существуют формы, или формализации, гарантирующие отсутствие изъянов в аргументах.

Надо полагать, в основе такого утверждения лежит то обстоятельство, что определенные виды формализации дают возможность формулировать связи между операторами в соответствии с доказательством с точки зрения правил обработки операторов.

Хотя в принципиальном плане это свойство чрезвычайно интересно, хотя оно обеспечивает необходимую основу для механического конструирования доказательств и верификации, хотя оно, наконец, иногда применяется людьми при обосновании тех или иных выводов, это свойство не гарантирует безошибочности используемых при разработке программного обеспечения аргументов, построенных на формализации. В сущности, вопрос о том, совместимы ли описания, используемые при разработке программного продукта, и элементы окружения, которые, как предполагается, моделируются с помощью этих описаний, в любом случае решается на основе интуиции. Несомненно, число ошибок в ходе такого моделирования в какой-то степени зависит от формы или языка описания. Однако выбор того или иного языка в любой ситуации зависит не только от степени формализации описания, но в равной мере и от характера моделируемого элемента окружающей среды, и от предшествующего опыта программиста. Поэтому те, кто утверждает, к примеру, что в процессе выбора той или иной формы символики при разработке программного обеспечения решающее значение имеют ее математические свойства, весьма недооценивает человеческую интуицию. Далее, следует констатировать, что те правила метода, которые предполагают обязательное использование программистом конкретных, ограниченных форм выражения, возможно, ведут к увеличению числа ошибок в программном продукте. Читателей, интересующихся более подробным анализом этого вопроса, мы отсылаем к [9].

Наконец, рассмотрение интуитивных ошибок в контексте третьей группы проблем, касающихся метода (порядок действий), показывает, что обязательный, заранее заданный порядок, пожалуй, не помогает избежать ошибок; более того, он может способствовать возникновению новых. В самом деле, в процессе разработки программного обеспечения программисту приходится делать интуитивные выводы исходя из сложных схем взаимосвязанных ограничений и требований, вытекающих из поставленных задач. Появление ошибок при этом гораздо менее вероятно, если разработчик получает возможность сосредоточиться на содержании этих сложных схем. Но такое условие вполне может быть нарушено в ситуации, когда порядок решения тех или иных вопросов навязывается программисту извне.

Подведем итоги нашей дискуссии о методах и интуитивных ошибках. Как представляется, применение того или иного метода может быть полезным главным образом постольку, поскольку он вооружает программиста контрольным списком обязательных действий и, тем самым, устраняет ошибки, возникающие "по забывчивости". С другой стороны, в сфере, касающейся определения форм выражения и порядка выполнения действий, польза от жестких ограничений, налагаемых методом, представляется довольно сомнительной.

Сделанный нами вывод вызывает новые вопросы, связанные с самой жгучей проблемой разработки программного обеспечения. Ведь при оценке того или иного метода дело заключается не только в том, какие преимущества может дать его использование, но и в том, как обеспечить

надлежащее соблюдение правил этого метода программистами, участвующими в том или ином проекте. Эту мысль можно сформулировать и в более общем плане: как побудить программистов надлежащим образом использовать подходящие к данному случаю и хорошо им известные приемы и наработки? Впрочем, тут же напрашивается еще один вопрос: в чем главная причина ошибок в программных системах: в том, что их разработчики не придерживаются определенного метода или, может быть, в том, что просто по халатности они не применяют, как это нужно, хорошо известные приемы?

Ответ, который, пожалуй, содержится уже в самом вопросе, в каком-то смысле подтверждается на примере другой широкой сферы созидательной деятельности человека — на примере строительства. В 1984 году в Дании было проведено исследование зданий, построенных в пятью годами ранее. В ходе него обнаружилось большое количество огрехов, большинство из которых в свое время легко можно было устранить с помощью хорошо известных и повсеместно распространенных приемов — но это не было сделано по причине халатности.

Этот пример подтверждает правильность сделанного нами ранее вывода о том, что проблема разработки высококачественного программного обеспечения не может быть решена с помощью правил и методов, которые в сущности приравнивают действия программиста к работе некой машины, производящей программы. Мы предлагаем здесь другую точку зрения на работу программиста: его главная задача состоит в том, чтобы строить теоретические модели решения стоящей перед ним проблемы с помощью компьютерной программы.

Но эта идея подробнее рассматривается в другой работе [10].

## Выводы

Главный вывод из настоящего выступления состоит в том, что разработка программного обеспечения во всех своих фазах и независимо от приемов, применяемых при ее реализации, должно и всегда будет строиться на интуиции.

Магистральный путь сокращения числа интуитивных ошибок состоит в многократном повторении операций и проверок. Решения о том, где и как применять правила, согласно которым ведется разработка программного обеспечения, принимаются интуитивно. Поэтому неверно полагать, будто разработка программного обеспечения сводится в основном к применению методов, состоящих из правил, и символики, ибо на самом деле более глубокая проблема разработки состоит в построении программистом теорий компьютерных (computer-based) решений.

И, наконец, вопрос по теме нашей конференции: какую же роль играет семантика в разработке программного обеспечения? Ответ: роль семантики нельзя сравнить с ролью композитора или либреттиста, с ролью дирижера, героя или героини. Семантика играет роль суфлера; она всего лишь подсказывает актерам реплики, которые им и без того известны, но просто не пришли на память в нужный момент.

## Литература

- [1] P. Medawar (1982) *Pluto's Republic* // Oxford University Press.
- [2] K. R. Popper (1959) *The Logic of Scientific Discovery* // Hutchinson, London.
- [3] W. v. O. Quine (1960) *Word and Object* // M.I.T. Press, Cambridge, Massachusetts.
- [4] A. N. Whitehead (1942) *Adventures of Ideas* // Pelican Books, Harmondsworth, Middlesex, England.
- [5] O. Jespersen (1933) *Essentials of English Grammar* // George Allen and Unwin, London.
- [6] M. Kline (1977) *Why the Professors Can't Teach* // St. Martin's Press, New York.
- [7] H. Ledgard, J. A. Whiteside, A. Singer, W. Seymour (1980) *The Natural Language of Interactive Systems* // *Communications of the ACM*, Vol.23, No.10, p.556-563.
- [8] B. Shaw (1946) *The Doctor's Dilemma* // Penguin Books, Harmondsworth, Middlesex, England.
- [9] P. Naur (1982) *Formalization in Program Development* // *BIT*, Vol.22, p.437-453.
- [10] P. Naur. *Programming as theory building* // *Microprocessing and Microprogramming*.