

В.М.Тавровский

ОТКУДА БЕРУТСЯ ЧАСТНЫЕ АЛГОРИТМЫ ДЕЙСТВИЙ ВРАЧА

Очерки разработчика

СОДЕРЖАНИЕ

ВВЕДЕНИЕ.

Часть 1. КАК ЭТО СЛУЧИЛОСЬ.

Случайность, не обещающая повторений.

Нежданное приближение к идее.

Осознание элементарной клинической ситуации.

Удивительные последствия.

Исчезнувшие предпосылки.

Откуда же берутся алгоритмы действий врача?

Часть 2. ИНСТРУМЕНТЫ И ПРИЁМЫ РАЗРАБОТЧИКА.

Оrientиры: группы алгоритмов, алгоритмы, классы ситуаций.

Принципиальная схема действий лечащего врача.

Сократовский метод.

Язык алгоритма.

Структурные элементы. Требование полного множества ответов. Параллели.

Комментарии к ответам. Рекомендации. Протокол рассуждений и выводов.

Рабочая гипотеза.

Типовые логические конструкции.

Элементарная клиническая ситуация. «Лечебные» и «специальные диагностические ситуации».

Другие «буквенные» ситуации и их комбинации. Циклы и повторы. Вставка-рекомендация и вставка-вопрос. Комбинированные конструкции.

Средства разработки автоматизированного варианта.

Файл-ситуация. Цикл «Ещё». Программа-редактор.

Пользовательская программа как инструмент разработчика.

Терминология.

Часть 3. ВЗАИМОДЕЙСТВИЕ С ЭКСПЕРТАМИ.

Что требуется от разработчика.

Кто может быть экспертом.

Общая схема взаимодействия.

Подготовка к сеансу работы.

Определение темы. Подготовка начальных эскизов. Запас свободных номеров.

Сеанс разработки.

Владение инициативой. Начальные вопросы. наброски типовых конструкций. Выбор на основе количественных, полуколичественных и качественных критериев. Комментарии к ответам. Рекомендации в лечебных ситуациях. Формулировки для оценки результатов лечения и наблюдения. наброски комбинированных структур во время сеанса. Консультации специалистов. Разрешение сомнений. Заключительная проверка.

После сеанса.

Сверхзадача разработчика.

Часть 4. РАЗВИТИЕ АЛГОРИТМА В ХОДЕ ЭКСПЛУАТАЦИИ.

Часть 5. ЧЕГО НЕ СЛЕДУЕТ ДЕЛАТЬ РАЗРАБОТЧИКУ.

Не создавайте без нужды новых сущностей.

Не надейтесь на память.

Не делайте ничего, что может задеть эксперта.

Не поступайте принципами и функциями автоматизированной системы.

ЗАКЛЮЧЕНИЕ.

ВВЕДЕНИЕ.

Термин «частные алгоритмы действий врача» я впервые употребил в нескольких публикациях 1978-1980 годов. Два смысла вкладывалось в него с самого начала: во-первых, идеальное представление о способе действий, о поведении, о структуре поведения лечащего врача, во-вторых, наименование инструмента, который можно создать и использовать для логически обоснованных и даже безупречных врачебных решений. Впоследствии я не раз писал о том и другом в статьях, монографиях и публикациях в Интернете, так что эти очерки рассчитаны на читателя, уже знакомого с темой. Не знакомому проще всего обратиться по адресу www.vmtavr2.narod.ru к выпускам 57-70 рассылки «Зачем и как автоматизировать лечебно-диагностический процесс». Там же можно познакомиться и с самими алгоритмами, и с соответствующей компьютерной программой (весь комплекс выложен для свободного использования).

Однако один аспект этой темы до сих пор не был детально представлен: не изложена методика разработки алгоритмов. Этими «очерками разработчика» я хотел бы ликвидировать оставшийся пробел. Они посвящены не обоснованию алгоритмизации, не содержанию алгоритмов, не результатам их применения, а способам и правилам их разработки. Думаю, это может быть интересно не только другим (предполагаемым) разработчикам, но и просто тому, кто вообще заинтересуется алгоритмами действий врача: известно ведь, что лучший способ понять, что заложено в предмете, - это узнать, как его создают. Кроме того, представление о том, откуда взялась сама идея вооружить врача алгоритмами, что породило её и какие условия этому способствовали, тоже может дать пищу для размышлений. С последнего и начнём.

ЧАСТЬ 1. КАК ЭТО СЛУЧИЛОСЬ.

Случайность, не обещающая повторений. Это именно *случилось*: всё главное произошло случайно и в одночасье. И потом на протяжении ряда лет повороты в развитии идеи определялись случайностями, которые следовали то одна за другой, то через огромные интервалы вплоть до последней, приключившейся через 37 лет от начала событий и подтолкнувшей меня к этим очеркам (об этом позднее). Счастливые случайности всегда любопытны: занятно и хочется понять, так ли уж они были ни с того, ни с сего. А ещё случайность события означает, что оно со всеми своими полезными последствиями может не повториться. Если оно не станет известным, а ценность его понятной, то всё канет в Лету и второго раза не будет.

К тому, вообще-то говоря, идёт, а я не всё сделал, чтобы этому воспрепятствовать. Не рассказал с толком, чувством, расстановкой, как разрабатывались алгоритмы действий

врача. Множество великолепных врачей из солидных учреждений Новокузнецка, Новосибирска, Тюмени, Перми, Барнаула, Красноярска, Улан-Удэ вложили в эти алгоритмы свой опыт и знания, но отображением этого богатства в виде алгоритмов занимался, к сожалению, лишь я один. Других желающих овладеть этим необычным ремеслом не было. Моим экспертам-клиницистам было вполне достаточно полученного с их помощью результата, а помощь состояла всего лишь в ответах на мои вопросы в ходе обычного диалога о знакомом каждому эксперту предмете, о его предмете. Как я потом выстраивал и распределял полученную информацию, выявлял пробелы, противоречия, неосознанные закономерности и неписанные правила, как придавал конечному результату форму, пригодную для использования в работе врача, - всё это было моей собственной кухней. Иначе говоря, на мне начиналось и заканчивалось понимание того, как создаётся алгоритм. Я никому этого не передал. Не написал я и никакой инструкции на эту тему: что толку писать, если некому читать.

Но вот и ещё одна, теперь уж последняя, случайность: вдруг появился продолжатель (как говорится, не прошло и сорока лет). В пору, когда я фактически отошёл от дел, из другого города, в результате случайного знакомства с моим сайтом в Интернете. Понял идею, заразился, атаковал ею окружающих, изъявил готовность немедленно приняться за дело, включил телефон, электронную почту и скайп, наконец, явился лично и потребовал, чтобы его научили. Я взялся.

Первые же попытки подтвердили известную истину: одно дело – создавать что-то самому, подчиняясь интуиции, чувству меры и гармонии, собственному вкусу, совсем иное – внятно объяснить, почему надо поступать именно так, а не иначе. То, что казалось само собой разумеющимся, приходится формулировать. Сначала возникают вопросы «что» и «как», на которые легко отвечать точно и безапелляционно, как и полагается учителю. Но затем следуют «почему», «для чего», «а нельзя ли иначе», а там и возражения, собственные предложения и даже самочинные действия. Подавлять эту активность нельзя, остаётся глядеть в оба, чтобы тебя правильно поняли, чтобы из твоих утверждений не сделали ошибочных выводов. Заодно и утверждения эти пересматривать, проверять на прочность. Предлагаемые очерки – побочный результат именно такой работы: с алгоритмами, с последователем, с самим собой.

Но вернёмся к истокам.

Нежданное приближение к идее. Был 1977-й год. Я работал над лекцией врачам-курсантам о тактике при округлых образованиях в лёгких. Содержание явно просилось в простую схему со стрелками: от симптомов – к предположениям, к способам выбора ра-

бочей гипотезы, от гипотез – к действиям. Схема была нарисована на ватмане, но первоначальное наведение порядка в рассуждениях тут же побудило к детализации. От основных гипотез (опухоль, туберкулёз, неспецифическое воспаление) нельзя было не спуститься на уровень ниже (рак или доброкачественная опухоль, туберкулома или туберкулёзный инфильтрат, абсцесс или паразитарная киста), нельзя было обойти вниманием и «соседние» области (лимфогрануломатоз, саркоидоз). От действий захотелось провести стрелки к результатам. От результатов, состоящих в ошибке или неудаче лечения, – к следующим гипотезам и следующим действиям. Одного ватмана для лекции не хватило, пришлось нарисовать переходящие друг в друга три схемы.

Лекция получилась очень предметной, известные общие соображения развивались в правила, курсанты строчили свои конспекты. Вывод следовал неизбежно: надо держать эти правила перед глазами не только на лекции, но и в повседневной клинической работе. Не тут-то было. Для принятия решений о конкретном больном и трёх схем оказалось мало. Потребовалось отвечать на множество дополнительных вопросов (о давности болезни и её динамике, возрасте пациента, его профессии, ряде симптомов, о согласии на операцию, о сопутствующих заболеваниях и проч.). Оказалось, что в лекциях я передаю курсантам не всё то, что надо иметь в виду при принятии решений, не всё то, что знаю и умею сам. Пришлось всё сделанное детализировать дальше.

Осознание элементарной клинической ситуации. Деталей оказалось много. Первоначальные три схемы пришлось дробить на связанные между собою фрагменты. О ватманах уже не могло быть и речи, их заменили сначала простые листы бумаги, а потом – карты с краевой перфорацией формата К4 (половина листа формата А4). Карты пришлось кстати потому, что краевая перфорация позволяет их кодировать, а потом по коду быстро извлекать из картотеки с помощью спиц. Таким образом, каждая деталь разукрупнённых схем получала свой код, связь одной детали с другими обозначалась на карте кодами этих других, а все детали собирались в картотеке, не вынуждая заботиться об их порядке благодаря кодированию.

Оставалось определить принцип детализации, правило, позволяющее устанавливать завершение одной детали и начало другой не на том основании, что на перфокарте просто не хватает места, а по смыслу работы врача с пациентом. Такое правило проявилось как-то само собой, из естественного представления о том, что врач сначала изучает ситуацию с помощью соответствующих моменту доступных средств, затем на этой основе выбирает способ действий (обычно из нескольких возможных), а затем ему необходимо время для осуществления сделанного выбора. Здесь в рассуждениях наступает пауза, вре-

мя используется на осуществление запланированных действий и наблюдение за получаемыми результатами. Надо, следовательно, указать разумную длительность этой паузы, а за нею наметить то, что может быть: все возможные результаты, за каждым из которых последуют свои логичные решения и действия.

Итак, на отдельной перфокарте умещалась следующая последовательность: осуществляемые к настоящему времени действия (лечебные или диагностические), сроки, отведенные для этих действий, полученные с их помощью сведения, необходимые для выбора дальнейших действий, перечень всех возможных альтернативных решений и аргументы для выбора, наконец, коды тех ситуаций, к которым выбранные решения ведут.

Эта последовательность повторялась от одной перфокарты к другой и вдруг однажды я сообразил, что держу перед собою структуру молекулы, из которой построено всё поведение лечащего врача. Это было совершенно неожиданно. Отсюда, из этого осознания последовало всё остальное. Изложенную последовательность (действия, дающие информацию, - перечень возможных способов дальнейшей работы с пациентом – аргументированный выбор из этого перечня – переход к реализации намеченного с назначением срока) - я назвал *элементарной клинической ситуацией*. - Чаще всего она умещалась на карте, иногда на ней удавалось совместить более одной ситуации, а иногда сложные рассуждения вынуждали к использованию двух карт, но всегда элементарная клиническая ситуация – это состояние, когда выполняют принятое ранее решение и к определённом сроку предстоит оценить новые результаты и принять новое решение.

Удивительные следствия. Случайное осознание сделанной находки сразу породило логичную систему понятий и соответствующих им терминов, систему, позволяющую лучше понять и описать врачебную работу вообще, как таковую. Переходящие друг в друга элементарные ситуации образуют *линии*. Движение по линии – это решение медицинской задачи по заранее описанной и обоснованной логике. Так как некоторые ситуации могут ставить перед врачом не одну задачу, а сразу две и более, то одновременно могут существовать несколько линий, *параллелей*. Параллели могут развиваться и завершаться независимо друг от друга, а могут противоречить друг другу (тогда одна из них закрывается в связи с *противоречием параллелей*) или дублировать друг друга (тогда используется *поглощение параллели*). Чем больше параллелей (относительно самостоятельных медицинских задач) возникает в ходе ведения пациента, тем сложнее работа врача. *Сложность пациента*, таким образом, может характеризоваться числом параллелей, из которых сложена клиническая картина.

Врач может использовать *отклонения от алгоритма*: что-то добавлять к его рекомендациям (*усложнение звена*) или что-то опускать (*упрощение звена*), пропускать одну из ситуаций (*сокращение цепи*) или, наоборот, повторить те же свои действия, хотя алгоритм этого рекомендует идти дальше (*удлинение цепи*). Как видно, перечисленные термины дифференцированно характеризуют индивидуализацию врачебной тактики, учёт таких конкретных обстоятельств, которые нельзя отразить (или ещё не успели отразить) с помощью правил.

Не менее удивительным, чем внезапное осознание поведения лекаря как последовательности элементарных клинических ситуаций, было восприятие алгоритмов врачами. Казалось бы, необычность предложенного им инструмента да ещё для обязательного использования должна была вызвать сопротивление, но его практически не было, ни явного, ни скрытого. Конечно, реплики вроде «Я и без алгоритма всё это знаю» были, но они прекращались, стоило только с помощью алгоритма показать оппоненту просчёты, недосмотры, а то и грубые ошибки в его собственной работе. Поразительно (или показательно?), что я не встретил непонимания со стороны своей ближайшей сотрудницы по кафедре – доцента Ларисы Петровны Чумаковой, человека с твердым характером и не менее твёрдыми клиническими правилами да ещё терапевта, тогда как я – хирург. «С души воротит от перфокарт, но понимаю, что это здорово» - под таким девизом она сама участвовала в разработке алгоритмов «Пульмонология» и «Химиотерапия воспаления», а также вместе с ассистентом Аркадием Львовичем Ханиным провела сравнительный анализ работы врачей клиники до алгоритмов и после их внедрения.

Алгоритмы были разработаны и немедленно введены в практику первоначально только «для себя», в клинике, которой я руководил. Выход за её пределы не был преднамеренным, он стал результатом нескольких случайностей. Сначала к моему нововведению проявил интерес профессор педиатрии Юрий Евгеньевич Малаховский (мы были с ним дружны и сопредседательствовали на городских клинико-анатомических конференциях). И уже очень скоро он сам и его доценты стали регулярно посещать меня для работы над алгоритмом «Педиатрия». Через несколько месяцев этот алгоритм был внедрён в детской клинической больнице № 4 Новокузнецка. Не меньшей случайностью был запрос на алгоритмы от главного врача больницы Новосибирского Академгородка Владислава Георгиевича Козлова. Его только что пригласили туда из Новокузнецка, так что он мельком видел мои алгоритмы и уловил, что их можно использовать для управления врачебной работой. Управление ему как раз и понадобилось на новом месте, где врачебный коллектив пребывал в состоянии разброда и шатаний (из-за чего и потребовалась звать на руководство варяга). Он поступил радикально: автобусом привёз в Новокузнецк всех заве-

дующих отделениями, показал им мои алгоритмы в действии и заразил идеей сделать то же самое у себя. Так последовали мои длительные командировки в Новосибирск. В результате контактов со специалистами этой больницы набор алгоритмов вскоре пополнился «Кардиологией», «Гастроэнтерологией» и «Акушерством и гинекологией». Между прочим, раздел «Роды и послеродовый период» последнего алгоритма, разработанный одним из первых после «Пульмонологии» именно в Академгородке с прекрасным специалистом Надеждой Николаевной Максимовой, оказался потом самым востребованным и долговечным в Новосибирске, Тюмени, Красноярске, Улан-Удэ и Барнауле.

Совсем уж неожиданным был визит руководителей Новокузнецкой станции скорой помощи Анатолия Захаровича Виноградова и Юрия Михайловича Янкина. Никогда никаких контактов не было у меня со скорой помощью, но они настаивали, что алгоритмы – это именно то, что им необходимо, и что вполне осуществимо в их специфических условиях. Результатом стало появление и немедленное внедрение алгоритмов для диспетчеров скорой помощи и врачей выездных бригад.

Последней случайностью была моя встреча с молодым заместителем заведующего Тюменским горздравотделом Александром Ивановичем Макаровым. Он ездил по городам и весям в поисках полезных организационных систем, а Новокузнецк в этом смысле тогда выделялся. Обо мне он услышал за несколько часов до своего поезда и зашёл, благо я задерживался на работе. Так началась наша многолетняя творческая дружба, ближайшим результатом которой в следующие два года стало внедрение алгоритмов в крупных учреждениях Тюмени: на станции скорой помощи, в многопрофильной городской больнице № 2, в детской больнице и в родильных домах.

Дальнейшее уже не было случайным. Сработала особенность станций скорой помощи: их руководители в разных городах общаются между собою, по крайней мере, так тогда было на востоке страны. Поэтому на протяжении 4-5 лет я помогал главным врачам вводить алгоритмы для скорой помощи в Новосибирске, Барнауле, Перми, Красноярске, Улан-Удэ и Владивостоке. Скорая помощь всегда на виду у руководителей городского здравоохранения. Через самых активных из них поветрие распространялось на другие учреждения. Способствовали этому и появившиеся публикации, и две конференции по скорой помощи союзного масштаба (в Новосибирске и Тюмени).

Не перестаю и сейчас удивляться этому процессу, никак не инициированному сверху, никак специально не организованному, этой спонтанной активности руководителей институтских клиник, городских лечебных учреждений, органов здравоохранения, среди которых я вдруг обрёл не только деятельных сторонников, но и личных друзей, в

разных городах и на долгие, как оказалось, годы (редкий подарок, если тебе уже серьёзно за пятьдесят).

Но ещё удивительнее, что под их руководством алгоритмы везде и сразу приживались и потом эксплуатировались годами, причём всё это было ещё до появления компьютеров, до автоматизации. Даже те организаторы здравоохранения и институтские профессора, которые вынужденно наблюдали необычные события на подведомственной им территории, проявляли скорее сдержанную заинтересованность, чем скепсис. Во всяком случае, противодействия со стороны власти имущих, за исключением трёх-четырёх слабых попыток, не было.

Моё единственное объяснение такому феномену состоит в том, что алгоритмы действий врача удовлетворяют объективно существующую потребность практической медицины. Волей-неволей подумаешь, всё ли сложилось так случайно, не было ли у этих событий серьёзной причины, существенных предпосылок. Конечно, предпосылки, благоприятные условия, почва, в которой внезапно раскрылось случайно попавшее в неё зерно, были.

Одно условие после только что приведенных описаний, напрашивается сразу: это свобода действия для руководителей медицинских учреждений и органов здравоохранения. В своей сфере ответственности они могли вводить организационные новшества, не испрашивая разрешения сверху. Главный врач тогда был более самостоятельной фигурой, чем сегодня, а хороший главный врач, пользующийся авторитетом в своём коллективе, нередко считался незаменимым, знал это и держался независимо. Это были зачастую инициативные и амбициозные люди, видевшие свою успешность в успешной деятельности своих учреждений, в доброй славе этих учреждений. О своих связях «наверху» они заботились лишь постольку, поскольку это было полезно для больницы, поликлиники или станции скорой помощи. К сожалению, социальные потрясения за 30 лет сказались на положении главного врача, на требованиях к нему, на степени его свободы. Амбициозности стало меньше, бюрократизация сделала инициативу наказуемой, финансовая зависимость вынуждает приспособляться к чиновникам. Так что одно из благоприятных условий для преобразований в организации врачебного дела зримо ослабевает, а, может быть, и вообще исчезает.

Конечно, относительная независимость главного врача – это условие внешнее, никак не связанное с сущностью алгоритмизации, оно частично объясняет реализацию, распространение и развитие идеи, а хотелось бы узнать о предпосылках её рождения. Были и такие, принципиально важные, личные и социальные. К сожалению, ныне тоже исчезнувшие.

Исчезнувшие предпосылки. Со студенческих лет меня привлекало представление различных процессов в виде схем, где квадратами с текстом обозначались последовательные этапы, а стрелками - переходы от причин к следствиям и превращения следствий в причины последующих событий. В далёком 1950 году профессор патофизиологии Днепропетровского мединститута Эммануил Яковлевич Стеркин с помощью таких схем завораживал студентов на своих лекциях, увлекался их логичностью сам и время от времени прерывал возбуждённое изложение коротким обращением к нам: «Понятно, да?». Да, было замечательно понятно.

Понятностью, логичностью, точным выражением мысли привлекали меня схемы распознавания и лечения острых отравлений, этапной помощи в военно-полевой хирургии, удивительные «Очерки гнойной хирургии» В.Ф.Войно-Ясенецкого, замечательные книги по оперативной хирургии, где действия хирурга расписывались с предельной детализацией. К порядку, последовательности, обоснованности, конечно, побуждала профессия хирурга и ранняя самостоятельность в этой сфере

Что касается ранней самостоятельности в хирургии, то она особенно побуждала опасаться ошибочных решений и поступков, пытаться их предвидеть, продумывать альтернативы. В начале 60-х мне попала редкая в своём роде книга «Ошибки, опасности и непредвиденные осложнения» Э.Гессе С.Гирголава и В.Шаака (1936). Впечатление от скрупулёзности в описании возможных ошибок было очень сильным, наложило отпечаток на всю мою деятельность в медицине. Моя докторская диссертация была наполовину посвящена ошибкам и опасностям в лёгочной хирургии, а потом это стало содержанием отдельной книги (в соавторстве с Е.А.Вагнером).

Этот мой интерес к пониманию и предупреждению ошибок, совпадающий со стремлением к упорядоченным рассуждениям, к логичным и чётким схемам для выбора решений, со временем только усиливался. Сыграло роль повышение личной ответственности, когда в семидесятых годах от заведывания отделением я сразу перешёл к руководству 400-кочной клиникой. Ответственность повысилась, а число врачебных недочётов в поле зрения возросло. Сюда же добавился опыт рецензирования и председательства на городских клинко-анатомических конференциях: я видел ошибки, видел, что большинство из них – это либо результат нелогичного рассуждения или досадных упущений памяти, что их могло бы не быть, но в то же время от них никто не застрахован, тем более, в условиях бурного роста медицинских знаний и возможностей. А рост продолжался, и вероятность неоптимальных решений в работе врача только увеличивалась. Этому надо было найти какое-то решение.

Рост знаний, рост информации в тех же семидесятих остро ощущался во всех областях знания, не только в медицине. Не за горами маячила эра компьютеров. А пока притчей во языцех стали системный подход, системный анализ, системотехника, обратные связи, управление, кибернетика, – и всё это иллюстрировалось схемами с квадратиками и стрелками («квадратология плюс стрелометание» – смеялись остряки).. Естественно, я был готов к восприятию этих понятий, к тому, чтобы с их помощью найти приёмы, хоть как-нибудь гарантирующие оптимальность врачебных решений.

Так возросшая личная озабоченность и личные пристрастия совпали с объективными проблемами быстро развивающейся медицины и с общим состоянием умов на подступах к информационной революции.

Я думаю, что теперь этих предпосылок нет, они исчезли. Вскоре после того, как компьютеры взяли на себя функцию запоминания, расчёты и отчёты, утихли и разговоры о системном подходе в медицине, а значит – и о логичности врачебных действий, о предотвращении неточностей и ошибок за счёт логичных и детальных схем. В качестве решения задачи на первый план выступили наперёд составленные медицинские стандарты и протоколы. Не инструмент логического вывода, не навигатор, обеспечивающий пошаговое движение с учётом конкретных обстоятельств, а одномоментно заданный набор действий при однажды названном диагнозе или состоянии. Здесь врачу не предлагают рассуждать. Здесь требуется подчинение стандарту, который кем-то составлен. В алгоритме, напротив, предлагается рассуждение: возможности выбора, аргументы для выбора, логичные следствия сделанного выбора, оценка результатов сделанного выбора. Это – естественный способ в медицине: так умудрённый опытом специалист помогает менее опытному, советует равному, обосновывает своё поведение перед старшим.

Медицина пропустила этот естественный для себя путь развития и вышла на простые, командные решение. А вот тут включилась новая сила: там, где правильность сводится к соблюдению команд, управление легко перехватывает чиновник. Чтобы следить за соблюдением стандартов, ни заведующий отделением, ни начмед, ни главный специалист, ни профессор медицины не нужны. Не нужны носители знаний и опыта. Простое решение найдено, закреплёно официально, объективная потребность этим паллиативом будто бы удовлетворена, голод утолён. И спонтанного возврата к естественному решению задачи не будет нигде.

Остаётся предполагать, что уже высказанная и реализованная однажды идея алгоритмизации медицинской практики, подкреплённая описанием её истории и изложением способов создания алгоритмов, вдруг да дожждётся своих энтузиастов, желающих с нею в руках противостоять общей тенденции. Опираясь на эту хрупкую надежду, продолжим.

Откуда же берутся алгоритмы действий врача? Известно, что строгий рецензент на клинико-анатомической конференции всегда укажет, что именно и почему надо было сделать не так, а иначе. Он это знает. И с ним согласятся другие, они тоже знают. Правда, вполне возможно, что, несмотря на знание, с ними случаются те же самые беды. Это справедливо и для других форм принципиального разбора врачебной тактики.

На дотошные вопросы разработчика алгоритмов опытный врач ответит всегда и предметно: как поступать при установленном и при сомнительном диагнозе, при успешном лечении и при неудаче, при наличии ресурсов и при их дефиците, при осложнениях и сопутствующих заболеваниях. Он всё это знает, сообразит, выдаст. Правда происходит это не тогда, когда минуты на счету, не на обходе в больнице, не на приёме в поликлинике, а в специально выбранное для встречи время. И не в вольном изложении на тему, а под жёстким напором подготовленного интервьюера. И потому это не значит, что в своей реальной работе он ничего из собственных возможностей не упустит. Само по себе знание не гарантирует, что оно непременно будет использовано в нужный момент.

Но оно существует. Представления об оптимальном поведении врача в конкретных и меняющихся условиях существуют объективно: в умах врачей, в виде неосознанного идеала, идеала целенаправленной деятельности в условиях неопределённости. Его нельзя изложить, не владея врачебной специальностью, не вникая в технологические закономерности лечебно-диагностического процесса и не заботясь о соблюдении правил логики. Оно есть только у опытных врачей. Но, не выраженное в словах, в сформулированных правилах «на все случаи жизни», оно чаще всего существует аморфно, в виде пресловутой «интуиции» и уже поэтому не всегда используется даже самими их обладателями. А ведь есть ещё и проблема огромных объёмов информации, которая наваливается на современного врача, из-за чего начинает действовать известный парадокс: богатство выбора затрудняет правильный выбор.

В повседневной работе врач не задумывается о своём положении в качестве основного звена системы «лечебное учреждение», не очень-то следит за логичностью своих действий и, конечно, ни один врач не владеет всей информацией даже в своей специальности так, чтобы не упустить ничего, относящегося к каждому моменту его работы с пациентом. Поэтому идеал не может реализоваться, врач движется поблизости от идеальной трассы, но не по ней.

Итак, идеальные представления о поведении врача существуют в умах, но в размытом, в раздробленном, в рассеянном виде. Их надо оттуда добыть и собрать. Алгоритмы действий врача – не сочинение разработчика, а его добыча: по зёрнам, по крупичкам, хотя

изредка попадают и россыпи, и самородки. Всё годится, всё надо собрать во всей полноте, всему этому богатству придать такую форму, чтобы сделать его действенным, доступным каждому лекарю на его рабочем месте. Разработчик – не автор, не сочинитель, он старатель. А потом ещё и ювелир, огранщик. Но не более.

Часть 2. ИНСТРУМЕНТЫ И ПРИЁМЫ РАЗРАБОТЧИКА.

Ориентиры: группы алгоритмов, алгоритмы, классы ситуаций. Алгоритм создаётся для врача на его рабочем месте. Основное содержание алгоритма – специфика этого рабочего места: симптомы, с которыми здесь имеют дело, способы обследования, средства лечения, логика и сроки принятия решений. Это специальность, но не вся специальность и не только она. Деятельность кардиолога различается в зависимости от того, работает он в стационаре, поликлинике или на скорой помощи. Кардиологам этих учреждений нужны разные алгоритмы, то же относится ко всем специальностям. Поэтому все алгоритмы делятся на три группы: для стационара, для поликлиники, для станции скорой помощи.

Внутри каждой группы каждый алгоритм, разумеется, ориентирован на определённую медицинскую специальность и часто именем этой специальности и называется. Здесь нужны две оговорки.

Первая состоит в том, что не разрабатываются отдельные алгоритмы для хирургических специальностей. Хирург имеет дело с теми же заболеваниями, которые до него и после него лечит терапевт, он пользуется теми же методами диагностики и терапевтического лечения и часто решает одинаковые с терапевтом задачи. С другой стороны, терапевт должен уметь определять момент, когда надо усомниться в дальнейших перспективах консервативного лечения и передать больного хирургу. Поэтому действия хирурга и терапевта описываются в одном и том же алгоритме (в «Кардиологии», «Пульмонологии», «Гастроэнтерологии» и т.п.).

Вторая оговорка касается алгоритмов, описывающих исключительно некоторые методы лечения, такие, которые усложнились настолько, что правильное пользование ими требует своих специфических знаний, а нужны они в разных областях. Таково использование антимикробных и других противовоспалительных средств, цитостатиков, разгрузочно-диетической терапии. Вероятно, сюда же можно отнести лучевую терапию, физиотерапию, анестезиологию (выбор и осуществление обезболивания), лечение синдрома диссеминированного внутрисосудистого свёртывания. Соответствующие алгоритмы представляют собою своеобразные приложения к алгоритмам для специалистов.

Итак, выбирая алгоритм для работы (или для разработки), достаточно исходить из рода занятий врача. Дальнейшая ориентировка основывается на состоянии пациента, ка-

ким оно представляется в данный момент при реальной встрече врача с больным или в воображении разработчика. В стационаре начальное представление чаще всего описывается бросающимися в глаза симптомами и синдромами: повышение давления, боль в груди, одышка (в практике кардиолога), боль в животе, рвота, понос (в гастроэнтерологии), снижение уровня гемоглобина, кровоточивость (в гематологии) и т.п. Бросающаяся в глаза первая картина сразу наталкивает на определённую группу предположений и вытекающих из них действий. Не важно, что предположения не всегда оправдываются: врачебная мысль получит необходимый импульс, а вызванные им действия, оцененные по алгоритму, быстро выявят ошибку, и тот же алгоритм выведет врача на верный путь.

Дальнейшие действия врача тоже определяются синдромами и их развитием. На этом основано разделение каждого алгоритма на «классы ситуаций». Определив ведущий синдром, врач уже знает с чего начать, из каких альтернатив надо выбрать первоначальные мероприятия. Именно к этому сводится содержание первых ситуаций каждого класса, отсюда, в зависимости от выбора, переходят в следующие ситуации.

Каждый специалист встречает и такие случаи, когда ясно, что пациент нуждается именно в его помощи, но ярко выраженных синдромов у него нет (неопределённые жалобы на сердце, немотивированный кашель, небольшие подъёмы температуры, невыразительные изменения лёгочного рисунка на рентгенограмме, небольшие отклонения от нормы в анализах). Такие ситуации в каждом алгоритме выделяются в класс «Дополнительный». Наконец, в алгоритмах для специальностей, где используются хирургические методы, создаётся класс «Послеоперационный период». Однако и эти добавления ещё не обеспечивают полного охвата деятельности врача алгоритмом.

Специалисту приходится учитывать, что у его пациента могут возникнуть заболевания, относящиеся к другим областям медицины. Общеврачебный минимум знаний обязывает его сориентироваться в таком заболевании и оказывать помощь до тех пор, пока сложность задачи действительно не потребует вмешательства другого специалиста или пока не подоспеет такой специалист. Это особенно важно, когда возникают угрожающие состояния. Чтобы содействовать врачу в действиях за пределами его узкой специальности в каждый алгоритм включаются ещё два класса: «Сопутствующие» и «Неотложные».

Изложенное разделение алгоритма на классы ситуаций относится к стационару. В алгоритмах для поликлиники (они, кстати, называются либо именем учреждения - «Детская поликлиника», «Женская консультация», «Тубдиспансер», либо именем специальности – «Амбулаторная кардиология») классы формируются иначе. Заботы врача поликлиники разделяются на две категории. Одна – это диспансерное наблюдение за взятыми на учёт пациентами (периодический контроль, профилактические мероприятия, лечение

обострений), другая – распознавание вновь выявленных заболеваний как у тех же, кто уже взят на учёт, так и у впервые пришедших в поликлинику. Отсюда и два класса ситуаций: «Группы учёта» и «Диагностика». В алгоритме «Тубдиспансер» жизнь потребовала создать ещё класс «Врачебно-трудовая экспертиза». Для исчерпывающей полноты всё это можно дополнить, как в стационаре, классами «Сопутствующие» и «Неотложные».

Алгоритм для диспетчера скорой помощи, принимающего вызовы, помогает в телефонном разговоре установить вид необходимой помощи и степень её срочности, а также придать этой информации формализованный вид, кодовое обозначение. Он разделяется на 4 класса, выбор между которыми ясен после ответа на первый вопрос: «Скорая слушает. Куда ехать?». Эти классы – «Вызов на дом или в детское учреждение», «Вызов на место работы пострадавшего», «Вызов в общественное место или на улицу», «Вызов из лечебного учреждения».

Алгоритм для диспетчера-эвакуатора служит для такого выбора посылаемой на вызов бригады, который согласуется с кодом, полученным от диспетчера 03, он же, когда такой свободной бригады нет, предлагает приоритеты для замены, а в зависимости от срочности – ещё и выбор между ожиданием и перенаправлением бригады с одного вызова на другой. Этот алгоритм разделяется на три класса: для центральной подстанции (она располагает полным набором специализированных бригад), для ближних подстанций (они могут пользоваться помощью Центра) и для подстанций отдалённых, вынужденных полагаться на собственные силы.

Наконец, для врачей выездных бригад нужны два алгоритма: «Общая помощь» (классы «Линейная бригада», «Линейная бригада – Дети», «Линейная бригада – Сердце, сосуды») и «Специализированная помощь» (классы «Реанимация», «Детская реанимация, «Сердце, сосуды», «Комы. Отравления. Коллапсы», «Травма», «Дыхание. Живот. Урология. Гинекология»). Помимо оказания помощи на месте событий эти алгоритмы программируют и другие решения, специфичные для скорой помощи на догоспитальном этапе (оставлении пациента дома или транспортировка в больницу, выбор больницы, помощь во время транспортировки, вызов бригады другого профиля на себя, планирование активного выезда к больному через несколько часов, передача информации об оказанной помощи участковому врачу).

Итак, алгоритмы, классы ситуаций, первая ситуация каждого класса, определяющая начало работы врача с пациентом, – таковы ориентиры в обширном пространстве практической деятельности врачей, в пространстве, описанном с помощью алгоритмов. С этих опознавательных точек начинаются как действия разработчика, так и, в реальной практике, действия врача.

Принципиальная схема действий лечащего врача. Разработчик должен постоянно видеть в работе лечащего врача чередование целенаправленных рассуждений и действий. Рассуждения необходимы для выбора действий, действия должны принести информацию для следующих рассуждений. Цель рассуждений – выбор действий, выбор из существующих возможностей. Эти возможности, полный их перечень, определяются ситуацией, в которой находятся врач и пациент, представлением врача о состоянии пациента в данный момент. Рассуждение, которое ведёт к выбору, – это оценка имеющейся информации с позиций медицинской науки и здравого смысла.

Действия предпринимаются в расчёте на получение определённого результата за определённый срок. Тем самым выбор решений сопряжён с чётким прогнозом, а их осуществление – с наблюдением за тем, как это прогноз подтверждается. Не позднее намеченного срока надо будет сопоставить результаты с прогнозом. Тогда сформируется новая ситуация, уточнённые представления врача о состоянии пациента, и это потребует принятия решений о следующем этапе действий.

Сократовский метод. Рассуждение врача на каждом этапе, в каждой клинической ситуации можно и нужно изложить в виде вопросов и ответов. Так как рассуждение должно обосновать выбор действий из перечня всех возможных альтернатив, то сначала надо, опираясь на доступную в данный момент информацию, уточнить состояние больного с расчётом сузить этот перечень, а затем сделать выбор из того, что осталось, и аргументировать его. Для такой задачи постановка вопросов в общем вида («Что это?», «Что делать?») непригодна. Вопросы должны быть предельно конкретными, требующими в ответ либо утверждения или отрицания («имеется / отсутствует / неизвестно»), либо выбора из заранее определённого перечня всех возможных решений.

Когда врач должен ответить на поставленный вопрос одним из всех возможных и заранее предложенных ответов, за которым последует очередной вопрос или вывод о необходимых действиях, это означает, что алгоритм, построенный с соблюдением логики, ведёт в рассуждении, организует его, направляет его. По сути, это сократовский метод.

«В современном варианте метод Сократа заключается в том, что свою мысль вы расчленяете на маленькие звенья, и каждую подаете в форме вопроса, подразумевающего короткий, простой и заранее предсказуемый ответ. По сути, это редуцированный, хорошо организованный [диалог](http://www.psychologos.ru/articles/view/psihologos) с перехватом инициативы». (Н.И.Козлов. Психологос.<http://www.psychologos.ru/articles/view/psihologos>).

Упомянутый перехват инициативы – не только важнейшее свойство алгоритма действий врача, способ превратить описания в предписания. Он ещё и необходимый приём во взаимодействии разработчика со специалистом-экспертом. Как надо пользоваться

этим приёмом, чтобы получить от эксперта именно те сведения, которые нужны для алгоритма, показано в третьей части очерков (в начале раздела «Сеанс разработки).

Язык алгоритма.

а) Структурные элементы. Язык алгоритма – это последовательность вопросов, ответов и рекомендаций. Каждому вопросу соответствует полное множество возможных ответов, из которых надо сделать обоснованный выбор. За ответом может следовать новый вопрос, а то и два, и более (или вопрос и рекомендация).

б) Требование полного множества ответов означает, что там, где предполагается ответить «да» или «нет», должен быть предусмотрен и вариант «неизвестно», а перечень содержательных ответов должен завершаться ответом «предпочтения затруднены». Такие конструкции не оставляют врачу тупиков: после ответов, не уменьшающих неопределённость, они направляют либо туда же, куда ведёт один из определённых ответов, либо на некий специальный путь, учитывающий неустранённую неопределённость.

в) Параллели. Когда ответ порождает одновременно более одного вопроса, рассуждение распараллеливается. Каждая параллель развивается дальше по своей логике. Параллели могут возникнуть и другим образом – в тех случаях, когда вопрос предполагает возможность множественного, а не единственного выбора. Для таких случаев используется так называемый цикл «Ещё». О нём – чуть ниже. При практическом применении алгоритмов параллели могут возникать и другим путём, по воле пользователя: он просто делает не один, а два захода в алгоритм (или более) из разных классов (например, при одновременно существующих двух болезнях).

г) Комментарии к ответам. К ответам можно (а часто необходимо) давать комментарии. Эти комментарии (они размещаются отдельно от ответов, не заслоняют их суть) несут важнейшую функцию: содержат аргументацию, необходимую для выбора данного ответа, поясняют необщеупотребительные или двусмысленные термины, могут напоминать о последствиях, связанные с данным выбором, ссылаться на официальные инструкции. Именно с их помощью алгоритмы подают врачу всё предметное богатство медицинских понятий с полной определённойностью. Комментарии к ответам не включаются в протокол хода рассуждений. Соответственно, их стиль может быть достаточно свободным, от него требуется только точность и выразительность, позволяющие врачу быстро ухватить необходимые аргументы.

Как правило, комментарии к разным ответам различаются, они ведь содержат аргументы для выбора. Но из этого формального правила есть исключение. Оно действует для ответов вида «имеются / отсутствуют», например, на общий вопрос о наличии какого-

либо состоянии: тогда перечень признаков этого состояния полезно поместить в комментариях к обоим ответам.

д) Рекомендации – третий элемент языка алгоритма – представляют собою решения врача о диагностических, лечебных и организационных мероприятиях. Это врачебные назначения, адресованные медицинской сестре, лабораториям, консультантам и самому себе. Это безапелляционные требования, которые необходимо выполнить, в результате чего предполагается и улучшить состояние больного, и получить новую информацию для последующих решений. Язык рекомендаций должен этому соответствовать: это язык чётких и недвусмысленных распоряжений. Приблизительность, оговорки, многовариантность указаний здесь нежелательны, их надо избегать. Рекомендации могут быть очень обширными, поэтому для удобства чтения во время работы с алгоритмом, часть их содержания можно вынести отдельно в виде комментария, который, в отличие от комментариев к ответам, будет включаться в протокол работы и никакой особой функции не несёт.

Благодаря рекомендациям алгоритм оказывается способным рационально воздействовать на использование времени в работе с пациентом: та рекомендация, которой заканчивается очередной этап этой работы, указывает на разумный срок, в пределах которого её надо выполнить и после этого двигаться дальше. В практике эксплуатации алгоритмов опоздание по отношению к заданному сроку означает дефект в работе врача или в его организационном обеспечении. опережение срока обычно вынуждено неожиданностями и чаще всего бывает при ухудшении в состоянии пациента.

е) Протокол рассуждений и выводов – один из важных результатов работы врача с алгоритмом в его автоматизированном варианте. Чтобы из вопросов и ответов он формировался в виде повествования, надо вопрос и ответы формулировать так, чтобы соединение вопроса с выбранным ответом образовывало утвердительную фразу, констатацию. Совокупность таких констатаций и образует протокол рассуждения, квалифицированное описание состояния пациента, обосновывающее намеченные действия. Такой протокол не только позволяет самому врачу уяснить положение вещей, но может служить и исчерпывающим докладом консультанту, контролёру, руководителю, другому врачу при передаче больного. Его главная особенность – чётко выраженная мысль врача.

С целью выразительности протокола надо вопросы писать заглавными буквами, а ответы – строчными; в обширных рекомендациях делать логически обоснованные абзацы, наименования медикаментов в назначениях выделять заглавными буквами.

Будущий протокол обязывает тщательно следить за орфографией, синтаксисом и пунктуацией, за чистотой и ясностью (без общих фраз, жаргонных выражений, канцеляризмов, необщепринятых аббревиатур и сокращений).

Рабочая гипотеза. В большинстве случаев логическая связка «Диагноз – лечение» не используется. На каждом этапе отправной точкой служит не диагноз, а его конкретные проявления: синдромы и симптомы, их динамика. Тем не менее, диагноз, конечно, всё время имеется в виду, но не в качестве установленной истины, а как предположение. В самом деле, диагноз далеко не всегда уже известен, он может оказаться ошибочным, сам по себе он ещё не определяет действия врача – важны его клинические проявления, те самые синдромы. Нередко необходим дифференциальный диагноз, требующий специальных усилий и времени, а воздействовать на состояние пациента нужно уже сейчас. Возможно и одновременное существование двух заболеваний, когда одним из них маскируется другое. Всё это означает, что в действительности врач некоторое время имеет в виду не диагноз, а диагностическое предположение, одну из нескольких возможных гипотез, выбранную в качестве руководства к действию.

В обычной практике недостаточная определённости с диагнозом нередко ведёт к неуверенным действиям. Не хочется ошибиться – и поэтому воздействуют сразу на две возможные болезни. Эмоционально тяжело признать грозное заболевание у человека во цвете лет, известить его об этом, приступить к тяжёлому лечению – давайте сначала попробуем то, что полегче. Неполнота или избыточность действий затушёвывают «классическое» течение болезни, время теряется.

Одно из ценных свойств алгоритмизации действий врача состоит в том, что описанные варианты событий исключаются. Рабочая гипотеза становится инструментом, последовательно уточняющим врачебную тактику. Во-первых, алгоритм предлагает врачу процедуру выбора рабочей гипотезы: весь перечень допустимых предположений и аргументы за каждое из них. При этом учитывается и такой вариант, когда врач затрудняется в выборе, - в этом случае алгоритм «сам» предлагает рациональный вариант. Во-вторых, выбор рабочей гипотезы означает точно соответствующую ей рекомендацию о действиях, способах контроля за течением болезни и контрольном сроке, не позднее которого надо проверить, оправдываются ли ожидания врача и подтверждается ли выбранная рабочая гипотеза. Строгое соблюдение при верной рабочей гипотезе обеспечивает максимально возможный успех, а при ошибочном предположении очень скоро вступает в явное противоречие с развитием событий. В последнем случае ближайшая этапная оценка полученных результатов обнаруживает отклонение от намеченной цели, и алгоритм предлагает пересмотреть либо лечение, либо рабочую гипотезу.

Типовые логические конструкции. Элементы языка алгоритмов – вопросы, ответы и рекомендации – позволяют составлять цепи рассуждений любой длины и сложности. В этих цепях обнаруживаются повторяющиеся конструкции, которые полезно знать, чтобы использовать преднамеренно и методично.

а) Элементарная клиническая ситуация. Это базовая конструкция алгоритма. Она позволяет описать наименьший логически завершённый этап принятия врачебного решения (вся работа лечащего врача не что иное, как последовательность этих этапов). Суть конструкции – решение конкретной этапной задачи. Она начинается рекомендацией о способах получения информации для решения этой задачи (врачебные мероприятия), а завершается исчерпывающим перечнем всех возможных решений. Середину конструкции занимает последовательность вопросов и ответов, которая, оперируя полученной информацией, ведёт к выбору нужного решения. Выбранное решение означает либо переход к следующему этапу работы, в следующую элементарную ситуацию, либо завершение работы с пациентом.

Представление об элементарной клинической ситуации необходимо для понимания алгоритма, как представление об атоме – для понимания материального мира. *Его надо отличать от термина «ситуация» без эпитета «элементарная»*, которым обозначают фрагмент алгоритма, ограниченный размерами перфокарты или пределами файла и обозначенный цифрой или буквой. Фрагмент этот тоже логичен, но может содержать не одну, а несколько переходящих друг в друга элементарных ситуаций. С другой стороны, длинная и ветвящаяся цепь рассуждения одной и той же элементарной ситуации может переходить из одного фрагмента в другой.

б) «Лечебная ситуация» и «Специальная диагностическая ситуация». Из всего разнообразия ситуаций выделяются две типичных конструкции: «лечебные ситуации» (для описания этапа лечения) и «диагностические ситуации» (для случаев использования высокоинформативных исследований – рентгеновских, эндоскопических, УЗИ, зондирований, ЭКГ и т.п.).

Лечебная ситуация излагает ограниченный временем этап лечения. Здесь в рекомендации перечисляются в определённом порядке необходимые врачебные назначения: диета, режим, виды медикаментозного лечения (препараты, дозы, ритм и способы введения), физиопроцедуры. Заключает рекомендацию абзац «Контроль», где перечисляются симптомы, за которыми надо следить, указываются способы лабораторного контроля и других специальных исследований, их периодичность, консультации.

Далее следует оценка результатов лечения по следующей схеме: фраза-вопрос «В пределах [установленного срока] симптомы...» и четыре ответа: «полностью исчезли»

(1), «уменьшились» (2), «сохраняются без перемен или «изменились противоречиво: одни исчезли, другие появились» (3), «усилились или появились новые» (4). Редакция вопроса и ответов варьирует в зависимости от медицинского содержания, отдельные симптомы могут обсуждаться детально (например, результаты лабораторных анализов, контрольной ЭКГ и проч.), но итогом всегда является один из четырёх выводов о результатах этапа: полный успех, улучшение, отсутствие значимых перемен или ухудшение. Возможен и 5-й вывод – особый вариант ухудшения (терминальное состояние, непереносимость препарата), но этим варианты исчерпываются.

Сам выбор одного из 4 или 5 выводов обеспечивается либо комментариями к ответам, либо деревом решений, которое может быть весьма сложным. Представление об этом можно получить из рисунка 1 на следующей странице. Это фрагмент файла-ситуации, который использует компьютерная программа. На рисунке видно, что при лечении пневмонии сначала оценивается рентгенологическая динамика. При этом третий и шестой ответ (в графе Dks) сразу ведут к результату 2 и 4 соответственно (в поле Kz). Остальные варианты рентгенологической динамики требуют, чтобы к ним добавилась оценка клинико-лабораторных симптомов.

При этом в случае первого, второго и четвёртого ответов о рентген-картине клинико-лабораторные симптомы оцениваются одним образом, в случае пятого ответа – другим образом, а при шестом ответе – третьим образом. Только после этого получается одна из четырёх конечных оценок. На том же рисунке видно (внизу), что алгоритм анализирует и возможное расхождение между динамикой рентгенологических симптомов и клинико-лабораторной картины. Объяснить такое расхождение можно трояким образом, для каждого из которых имеется свой выход. Эти выходы обозначены цифрами 7, 8 и 9. Понятно, что после выходов 8 и 9, кроме основной линии поведения, появится параллель для сопутствующего заболевания

Специальная диагностическая ситуация описывает всё, что связано с инструментальными исследованиями. В рекомендации, если она нужна, излагается подготовка к процедуре. Рассуждение отвечает на вопрос о тех результатах исследования, которые существенны для дальнейших решений. Для этого предлагается полный перечень таких результатов, вообще свойственных данному методу. Этому перечню предшествуют два стандартных ответа: «существенных изменений не найдено» и «исследование не удалось». Редакция в обоих случаях может варьировать, а сами эти ответы - детализироваться; в первом случае, например, возможны «норма», «несущественные изменения», во втором – «не удалось», «противопоказано», «пациент отказался от исследования».

V0310111									
	Ks	Tip	Dks	Kss	Kz	Smin	Smax	Ed	Text
	004	q		005					РЕНТГЕНОЛОГИЧЕСКАЯ ДИНАМИКА
	005	a	1	006					отсутствует, инфильтративных теней нет.
	005	a	2	006					- улучшение, инфильтративные тени исчезли.
	005	a	3		2				- улучшение, инфильтрат менее 1 см в диаметре.
	005	a	4	006					- отсутствует, инфильтрат менее 1 см в диаметре.
	005	a	5	008					- улучшение, инфильтрат не менее 1 см. в диаметре.
	005	a	6	010					- существенного улучшения нет, инфильтрат не уменьшился.
	005	a	7		4				- явное ухудшение.
	006	q		007					КЛИНИКО-ЛАБОРАТОРНЫЕ СИМПТОМЫ ВОСПАЛЕНИЯ
	007	a	1		1				отсутствовали с начала этапа.
	007	a	2		1				исчезли.
	007	a	3		2				уменьшились.
	007	a	4	012					сохраняются без перемен и выражены умеренно.
	007	a	5	012					появились.
	007	a	6	012					усилились.
	008	q		009					КЛИНИКО-ЛАБОРАТОРНЫЕ СИМПТОМЫ ВОСПАЛЕНИЯ
	009	a	1		2				отсутствовали с начала этапа.
	009	a	2		2				исчезли.
	009	a	3		2				уменьшились.
	009	a	4		2				сохраняются без перемен и выражены умеренно.
	009	a	5	012					появились.
	009	a	6	012					усилились.
	010	q		011					КЛИНИКО-ЛАБОРАТОРНЫЕ СИМПТОМЫ ВОСПАЛЕНИЯ
	011	a	1		3				отсутствовали с начала этапа.
	011	a	2		2				исчезли.
	011	a	3		2				уменьшились.
	011	a	4		3				сохраняются без перемен и выражены умеренно.
	011	a	5		4				появились.
	011	a	6		4				усилились.
	012	q		013					ОБЪЯСНИТЬ КЛИНИЧЕСКИЕ СИМПТОМЫ СОПУТСТВУЮЩИМ ЗАБОЛЕВАНИЕМ
	013	a	1		7				нельзя.
	013	a	2		8				можно, имеется сопутствующее легочное заболевание.
	013	a	3		9				можно, имеется сопутствующее внелегочное заболевание.

Рис.1 . Пульмонология. Фрагмент лечебной ситуации. Дерево решений приводит к одному из 4 типовых выводов (1-4), а при неожиданном появлении новых симптомов – к их вариантам (7-9).
Файл читается сверху. Вопросы и ответы помечены в поле Tip соответственно буквами q и a . В поле Kss указан код следующей строки (Ks): со строки-вопроса – это все возможные ответы, со строки ответа – это очередной вопрос. В поле Dks пронумерованы ответы на один вопрос. В поле Kz – код выхода из ситуации.

Общее свойство лечебной и специальной диагностической ситуаций состоит в том, что они требуются в своём неизменном виде в разных местах алгоритма, осуществляются по своей внутренней логике, а полученный результат используется в каждом месте по-разному, в зависимости от логики этого места. Естественное желание – каждую из них делать в единственном экземпляре и вызывать потом туда, где она понадобится. Это соответствует тому, что программисты называют функцией, такой подпрограмме, которая получает задание (информацию) из любого места программы, выполняет это задание и возвращает результат туда, откуда пришёл запрос.

Ситуации-функции, позволяющие одни и те же действия осуществлять в разных местах алгоритма, надо обозначать не числовыми номерами, как все прочие ситуации, а буквами. В своих алгоритмах я использовал буквы кириллицы и термин «буквенная ситу-

ация» в противовес «цифровой». Таким образом, однажды созданная "буквенная" ситуация может включаться сколько угодно раз в "цифровые".

Выходы же из такой «буквенной» (в отличие от остальных «цифровых») надо пронумеровать, начиная с единицы. Тогда «цифровая» ситуация 28 вызывает на себя «буквенную» ситуацию К (запись будет выглядеть, как "28-К"), второй результат «буквенной» приписывается к этой записи («28-К2-»), а дальнейшие действия (очередной вопрос или переход в другую «цифровую» ситуацию) определяются условиями и задачей ситуации 28.

в) Другие «буквенные» ситуации. Итак, буквенные ситуации – это выделенные процедуры, которые, будучи однажды созданы, могут включаться в разные места алгоритма, в разные цифровые ситуации. Кроме лечебных курсов и специальных диагностических процедур, ими можно описывать и некоторые другие задачи врача, возникающие в разных цифровых ситуациях. В качестве примера можно привести действия при возникновении показаний к хирургическому лечению, когда надо установить противопоказания, предложить операцию и получить согласие, преодолеть отказ, а после операции перейти к послеоперационному ведению. Другой пример – периодическая проверка сердцебиения плода в процессе родов. От результатов здесь зависят как лечебные меры в отношении плода, так и тактика акушера в ведении самих родов, причём и меры эти и тактика на разных этапах родов различаются. Третий пример – периодическая оценка состояния пациента на основе нескольких строго определённых признаков, каждый из которых может иметь 2, 3 или 4 существенных значения. Такая оценка необходима, например, при лечении кардиального шока, когда надо оценивать артериальное давление (повышено, нормальное, снижено), отёк лёгких (имеется, отсутствует, появился, усилился, уменьшился) и аритмию (имеется, отсутствует). Та же задача стоит при балльной оценке состояния новорожденного по Апгар. Число сочетаний в таких случаях составляет под сотню. Буквенная ситуация позволяет все их представить, зафиксировать, объединить в группы, ведущие к одинаковым мерам и передать это в цифровую ситуацию.

Обозначения лечебного курса или диагностической процедуры одной буквой ведёт к тому, что буква ассоциируется с содержанием ситуации. Стандартная нумерация всех или некоторых выходов из буквенной ситуации тоже легко запоминается: во всех специальных диагностических ситуациях «1» означает неудачу, во всех лечебных «1» - это полный успех, «2» - улучшение, «3» - безрезультатность, «4» - ухудшение.

Такая компактность и выразительность позволяют в пределах одной цифровой ситуации отразить сложные варианты, когда одни и те же действия приходится повторять в

разных сочетаниях. В работе врача с пациентом они не редки. Курс лечения приходится повторять или переходить к усиленному курсу, а потом возвращаться к предыдущему. Диагностические процедуры, процедуру получения согласия на операцию и преодоления отказа от операции приходится повторять и чередовать их с лечебными курсами. Переходы в таких комбинациях определяются результатом, который получается в каждой из них. Все это можно легко и наглядно изложить в виде комбинаций буквенных ситуаций с использованием трёх приёмов; циклов, повторов и вставок.

г) Циклы и повторы. *Цикл* – это такая конструкция, в которой один из конечных ответов возвращает на ту строку, которая уже была однажды пройдена. Пример *простого цикла* – когда выход «улучшение» в лечебной ситуации возвращает к тому же лечебному курсу, к той строке цифровой ситуации, на которой этот курс назначается. Такое возвращение может быть и не сразу, а после одного или нескольких других этапов – тогда это будет *сложный цикл*. В отличие от цикла, *повтор* – это последовательность из двух одинаковых буквенных ситуаций, вызванных с разных строк. Этот приём позволяет в первый раз оценить выходы из буквенной ситуации одним образом, а во второй раз – другим. В лечебной ситуации он без дополнительных вопросов учитывает длительность лечения (число курсов), чем нередко определяются следующие решения.

д) Вставка-рекомендация и вставка-вопрос. Между буквенными ситуациями, образующими описанные выше комбинации, можно делать вставки для указания на некоторые дополнительные действия. Так, после улучшения в лечебной ситуации, прежде чем повторить тот же курс, можно вставить рекомендацию об отмене части лечебных мер (например, об отмене антибиотиков), а при безрезультатном итоге – вставку с рекомендацией о дополнении в лечение. Вставка может заключать и вопрос, например, об общей длительности лечения, и уже от ответа на него, указывать на тот или иной дальнейший шаг. Возможна и комбинация вставки-рекомендации с вставкой-вопросом: так, можно рекомендовать консультацию того или иного специалиста или даже консилиум (так, например, приходится поступать при трудностях с распознаванием причины коматозного состояния), а потом спрашивать, какое из возможных решений было ими предложено.

е) Примеры комбинированных конструкций.

Предположим, что есть ситуация 17, вызывающая специальную диагностическую ситуацию Е: «17-Е-». Исследование не удалось: «17-Е1-». Теперь надо провести трёхдневную подготовку и повторить исследование:

«17-Е1- Готовьте к повторному исследованию 3 дня [даются рекомендации] –Е-».

Последнее «Е» - это не то же самое место, что было в первый раз, потому что тогда при неудаче получался бы порочный круг. Но это опять ситуация Е, только выход «1» из неё будет оцениваться уже иначе, например, при повторной неудаче надо применить исследование Ж: **«17-Е1-Готовьте к повторению- Е1-Ж-»**.

Очень полезны подобные комбинации для изложения сложной лечебной тактики. Допустим, есть три вида недельного курса лечения: основной (К), с которого начинают, облегчённый (Л) и усиленный (М). Действия излагаются в ситуации 28: **«28-К-»**. При ухудшении надо усилить лечение: **«28-К4-М-»**. Если через неделю станет лучше, полагается вернуться к основному курсу, а при дальнейшем улучшении – к облегчённому: **«28-К4-М2=К2-Л-»**. Облегчённое лечение надо проводить до полного исчезновения симптомов: **«28-К4-М2=К2-Л2-Л1-закончить»**. При этом облегчённый курс (Л) должен проводиться не менее 2 недель даже при самом благоприятном обороте: **«28-К4-М2=К2-Л1-Л1-закончить»**. Но если ухудшения за время лечения не было, то курс Л при его полном успехе повторять не надо: **«28-М2=К2-Л1-закончить»**.

Оставляю желающим удовольствие самостоятельно конструировать разные варианты на этом примере, а в заключение подчеркну очень существенное свойство подобных конструкций: они позволяют включить в решение проблем фактор времени. В ряде случаев, когда достигнут полный результат, лечение полагается ещё продолжить, закрепить. Для других случаев, когда такого результата нет, но достигнута стабилизация, надёжность последней нельзя устанавливать при первом курсе с результатом «без перемен», полагается продлить то же самое лечение в виде повторения таких же курсов до истечения определённого срока. Всё это предусматривает подобная комбинированная конструкция, а сверх того - и неожиданности при этих продлениях: ухудшение, подтвердившее ненадёжность стабилизации, и нежданное улучшение, обнаруживающее возможность ещё сдвинуть течение болезни к лучшему.

Средства разработки автоматизированного варианта. Алгоритмы разрабатывались и распространялись до эпохи компьютеров и даже с появлением последних – без них. В 90-х годах один программист попытался автоматизировать то, что я сделал на перфокартах, но потерпел неудачу. Самого меня подвигла на это опять-таки случайность, причём только в 2001 году. В вычислительном центре Кировской областной больницы, где эксплуатировались мои АРМы врача, появился практикант - толковый студент-дипломник. Меня попросили дать ему оригинальную тему. Я и предложил автоматизацию частных алгоритмов. Студент регулярно ходил ко мне, выслушивал мои объяснения и

представления о том, как подступиться к задаче, много и не всегда понятно для меня теоретизировал, выдвигал идеи, но за довольно большой срок так ничего и не сделал, хотя диплом на эту тему написал и успешно защитил. Но я к тому времени столько наобъяснял, что сам всё понял, и так разозлился, что сам сделал автоматизированный вариант: базу знаний, базу данных, программу для врача, а заодно и программу-редактор для разработчика. С этого момента стало ясно, что возврата к перфокартам не будет.

Автоматизация открыла новые возможности, но представленные выше средства и приёмы разработки остались актуальными. Они не поменялись, а пополнились.

а) Файл-ситуация. Фрагмент алгоритма, который был выше определён и назван «ситуацией», теперь помещён в отдельный файл и назван «файлом-ситуацией». Каждая строка этого файла-ситуации помечается в поле `tip` как вопрос, ответ или рекомендация. Содержание записывается в поле `text`, комментарии к ответам и рекомендациям – в поле `comm`. В строках для рекомендаций доступны поля для указания сроков.

Строки нумеруются, в специальных полях перехода указывается на какую следующую строку (или строки) либо в какой другой файл-ситуацию надо отсюда перейти.

Имя файла-ситуации состоит из номера алгоритма, номера класса и номера самого файла в этом классе. Для буквенных ситуаций имя завершается не номером, а CHR-кодом буквы. В специальном файле перечислены все алгоритмы и классы с их номерами. Опираясь на эту информацию, программа выбирает запрошенные алгоритмы, задаёт вопросы, воспринимает ответы, выдаёт рекомендации, намечает время следующего сеанса работы, печатает протокол и всё запоминает в базе данных.

Заполнение и редактирование файла-ситуации возможно только с помощью программы-редактора.

б) Цикл «Ещё». Эта конструкция используется в автоматизированном варианте алгоритма и служит для тех случаев, когда из предлагаемого перечня ответов можно выбрать не один единственный, а любое их число. Пример – специальные диагностические процедуры, которые могут дать сразу несколько результатов, каждый из которых требует к себе особого внимания врача. Другой пример – ответы на вопрос о ведущем синдроме (их может быть два и более) или о поражённой системе органов в классе «Сопутствующие» (их может быть несколько).

Конструкция представлена на рисунке 2. На вопрос о результатах посева мокроты можно выбрать один из 10 утверждающих ответов (они пронумерованы в поле `Dks`). Ответ запомнится для дальнейших действий в «цифровой» ситуации (его код стоит в поле `Kz`), но сначала надо выполнить требование в поле `Kss` – уйти на строку с `Ks=004` и задать вопрос «Ещё результат», а затем снова сделать выбор, на этот раз с возможностью сказать

«отсутствует». Процедура повторяется, пока не будет использована эта возможность или не будут выбраны все утвердительные ответы. В цифровой ситуации будут заготовлены параллели для всех утвердительных ответов. Их может быть только 6 (номера в полях Kz), а не 10, потому что некоторые из результатов на деле обязывают к одинаковым действиям.

V0312102									
	Ks	Tip	Dks	Kss	Kz	Smin	Smax	Ed	Text
	000	r		001					Б. Бак. и цитологические исследования при неясном диагнозе.
	001	r		002		1		w	ПОСЕВЫ МОКРОТЫ или промывных вод бронхов на МБТ (3 раза подряд).
	002	q		003					РЕЗУЛЬТАТ ОБСЛЕДОВАНИЯ
	003	a	#		7				отсутствует.
	003	a	1	004	1				- существенных для диагностики данных не получено.
	003	a	2	004	2				- рост МБТ при посеве.
	003	a	3	004	2				- положительный результат мазка мокроты на кислотоустойчивые бактерии.
	003	a	4	005	004				- сомнительный результат мазка мокроты на кислотоустойчивые бактерии.
	003	a	5	004	4				- обнаружены опухолевые клетки.
	003	a	6	004	5				- рост флоры из мокроты в диагностических количествах.
	003	a	7	004	5				- положительный посев крови на гемокультуру.
	003	a	8	004	6				- элементы грибов при микроскопии мокроты.
	003	a	9	004	6				- элементы грибов при микроскопии бронхоальвеолярной лаважной жидкости.
	003	a	10	004	6				- массивный рост грибов из мокроты.
	004	q		003					ЕЩЕ РЕЗУЛЬТАТ
	005	r		006		2	3	d	ПОВТОРИТЕ МАЗОК МОКРОТЫ НА КИСЛОУСТОЙЧИВЫЕ БАКТЕРИИ.
	006	q		007					В ПОВТОРНОМ МАЗКЕ МОКРОТЫ
	007	a	1		7				кислотоустойчивые бактерии не обнаружены.
	007	a	2		7				- сомнительный результат мазка мокроты на кислотоустойчивые бактерии.
	007	a	3		2				- положительный результат мазка мокроты на кислотоустойчивые бактерии.

Рис. 2. Цикл «Ещё» в диагностической ситуации. Каждый положительный ответ приводит не только к планированию дальнейших действий (код в поле Kz), но и к вопросу на строке с Kss=004 с необходимостью ответить на него словом «отсутствует» или ещё одним положительным ответом.

в) Программа-редактор алгоритмов действий врача. Основное назначение этой программы – исключить формальные ошибки при заполнении файла-ситуации. Она позволяет создать новый файл с правильным наименованием и редактировать имеющиеся с использованием всех изложенных выше конструкций. Соблюдение правильной последовательности вопросов, ответов и рекомендаций обеспечивается тем, что строки каждого вида вводятся программно, нажатием функциональных клавиш, причём нумерация строк и связи между строками устанавливаются автоматически (а потом могут корректироваться вручную). Одним нажатием вводится несколько дополнительных строк для ответов на вопрос, удаляются строки, оставшиеся не заполненными, создаётся параллель, делается заготовка для цикла «Ещё». Всё это, конечно, облегчает работу. Знание структуры файла позволяет легко его читать: проследить логику рассуждения при разных ответах и увидеть все переходы в другие файлы.

Буквенная ситуация, к которой обращается ситуация цифровая, используется точно так, как вообще в программировании используется *процедура*: она выполняет свою задачу

и передаёт результат (в нашем случае порядковый номер ответа) туда, откуда вызвана. Здесь для каждого ответа указываются соответствующие ему переходы к новым строкам с вопросами либо рекомендациями или выходы в другие файлы-ситуации.

Работа врача в автоматизированном варианте регистрируется в виде протокола «Ход рассуждения». Это предъявляет строгие требования к формулировкам вопросов, ответов, рекомендаций и комментариев. Эти требования были описаны ранее, но практическое значение они приобретают именно в автоматизированном варианте. Разработчик должен всегда представлять, как введенные им тексты будут выглядеть на экране и в напечатанном протоколе.

Наконец, в редакторе есть специальная функция проверки правильности заполненных файлов-ситуаций. Она позволяет в одном файле или сразу в целой группе выявить упущения и неправильности. Такие файлы выводятся на экран с указанием на ошибку и с предложением тут же её исправить.

г) Пользовательская программа как инструмент разработчика. Программа для врача полезна разработчику тем, что позволяет сразу посмотреть, как им сделанное будет выглядеть в процессе эксплуатации. Функция «Демонстрационный вариант» позволяет начать проверку с любого файла-ситуации (хотя можно придумать пациента с именем и фамилией, выбрать алгоритм и его класс и начать «работу» с первой ситуации класса).

На любом этапе можно получить протокол пройденного пути на экране, сохранить его в текстовом файле или напечатать. Такой протокол – отличный материал для того, чтобы проверить логику на её правильность, а тексты – на их полноту, отсутствие излишеств и культуру.

Терминология. Являясь специфической областью знания, алгоритмы действия врача и их разработка неизбежно оснащаются собственной терминологией. Термины – тоже инструмент, они необходимы для полного понимания своего дела, для точного обмена информацией между разработчиками и между разработчиком и экспертом, который тоже вынужден оперировать новыми терминами. Здесь совершенно необходимо единое словопотребление. Поэтому имеет смысл, пренебрегая некоторыми повторами, привести здесь небольшой терминологический словарь.

Алгоритм – сфера практических действий врача, ограниченная его специальностью и местом работы.

Класс ситуаций – раздел алгоритма, выделенный на основании таких признаков, с помощью которых врачи с самого начала способны предположить группу состояний или заболеваний, к которой относится пациент.

Элементарная клиническая ситуация – наименьший логически завершённый этап принятия врачебного решения, который включает в себя определённую цель, способы получения информации, перечень всех возможных решений задачи и рассуждение, позволяющее сделать выбор на основе полученных данных.

Файл-ситуация (вариант: ситуация) – фрагмент алгоритма, изложенный в файле и снабженный уникальным именем, в которое входят номер алгоритма, номер класса и код самой ситуации).

Вопрос – структурный элемент ситуации, за которым должно следовать не менее 2 ответов, и составляющий первую часть констатирующей фразы в протоколе «Ход рассуждения».

Ответ – структурный элемент ситуации, следующий за вопросом и составляющий окончание констатирующей фразы в протоколе «Ход рассуждения».

Комментарий к ответу – текст, поясняющий ответ или содержащий аргументы для его выбора.

Рекомендация – структурный элемент ситуации, содержанием которого являются врачебные назначения.

Комментарий к рекомендации – продолжение рекомендации.

Срок рекомендации – указанный в файле-ситуации срок, не позднее которого следует выполнить рекомендацию и вернуться к движению по алгоритму (в процессе работы пользователь может корректировать этот срок для конкретного случая)..

Лечебная ситуация – ситуация, содержащая рекомендацию о лечении и процедуру обоснования выбора между четырьмя основными результатами лечения: полным успехом, улучшением, отсутствием перемен и ухудшением.

Специальная диагностическая ситуация – ситуация, содержащая условия проведения диагностической процедуры, перечень всех возможных значимых результатов исследования и способ выбора одного или нескольких из них.

Цифровая ситуация – ситуация, код которой (последние 3 символа имени файла) является числом.

Буквенная ситуация - ситуация, код которой (последние 3 символа имени файла) является CHR-кодом буквы, а сама она играет роль процедуры, то есть вызывается из цифровой ситуации, выполняет свою задачу и возвращает результат туда, откуда была вызван.

Линия – последовательность вопросов, ответов, рекомендаций и переходов из одной ситуации в другую.

Заккрытие линии – программное логичное завершение линии, отмеченное специальным знаком в поле для перехода в другой файл.

Этап – отрезок линии от начала до ближайшего срока рекомендации или между двумя соседними сроками, или от последнего срока рекомендации до закрытия линии.

Параллели – одновременно существующие (не закрытые) линии.

Цикл - логическая конструкция в пределах одного файла-ситуации или в совокупности двух и более файлов, при которой один из ответов на вопрос в ходе диалога возвращает к одному из предшествующих участков рассуждения (к вопросу или рекомендации).

Цикл «Ещё» - логическая конструкция для множественного выбора из перечня ответов, в которой параллельно с каждым ответом, если это не ответ «отсутствует», повторяется вопрос «Ещё» и вновь предлагаются ещё не выбранные ответы.

Протокол «Ход рассуждения» – текст, автоматически сформированный в процессе работы с пользователем из задействованных им вопросов, ответов и рекомендаций.

К этому следует добавить термины, которые в разработке не нужны, а используются только при эксплуатации алгоритмов, - разработчик должен вооружить ими пользователя.

Произвольное закрытие линии – прекращение действий в связи с реальными обстоятельствами, делающие их продолжение невозможным (отказ пациента, самовольный уход пациента, противопоказания к лечению, летальный исход), отмеченное самим пользователем в поле для перехода в другой файл (возникает только в ходе работы пользователя).

Поглощение параллелью – произвольное закрытие одной линии в связи с тем, что предусмотренные ею действия уже осуществляются на другой (возникает только в ходе работы пользователя).

Противоречие параллели - произвольное закрытие одной линии в связи с тем, что предусмотренные ею действия запрещаются на другой (возникает только в ходе работы пользователя).

Отмена шагов – движение назад к одной из предыдущих развилок (но в пределах этапа) с тем, чтобы выбрать другой ответ (отмененные шаги не запоминаются). Осуществляется с помощью функциональной клавиши F2.

Пересмотр - движение назад за пределы этапа с тем, чтобы выбрать другой ответ в связи с признанием ошибки рассуждении (прежние действия запоминаются). Осуществляется с помощью функциональной клавиши F2.

Коррекция - движение назад за пределы этапа с тем, чтобы выбрать другой ответ в связи с получением новой, не полученной вовремя информации (прежние действия записываются). Осуществляется с помощью функциональной клавиши F2.

Укорочение линии – невыполнение рекомендации и игнорирование её срока (немедленное возвращение к движению по алгоритму).

Упрощение звена – невыполнение части рекомендации (программно не регистрируется).

Усложнение звена – дополнение рекомендации другими врачебными назначениями (программно не регистрируется).

Часть 3. ВЗАИМОДЕЙСТВИЕ С ЭКСПЕРТАМИ.

Алгоритм нельзя сочинить по данным литературы или курса лекций. Это полезные источники предварительной или дополнительной информации, но сам продукт рождается только в специально организованном диалоге разработчика с врачом-специалистом. Задача первого – извлечь знания, которыми располагает специалист, и придать им структуру алгоритма (вопросы, ответы, рекомендации), обеспечив логичность и логическую полноту изложения. Задача специалиста – выдать знания, следить за правильностью и содержательной полнотой возникающего дерева решений и по ходу работы время от времени давать экспертную оценку формирующегося результата. Не все могут с успехом быть участниками такого диалога.

Что требуется от разработчика. Разработчику полезно чувствовать себя участником тотальной алгоритмизации практической медицины. В каждый данный момент он развивает (расширяет, углубляет, модернизирует) то, что уже создано, опирается на него и отталкивается от него.

В этом смысле надо учесть и то, что сделано автором этих очерков. Во-первых, это перечень тех алгоритмов (их наименований), которые либо были сделаны и использованы в прошлом, либо лишь намечены. По существу, это перечень врачебных специальностей, для многих из которых указаны и их классы ситуаций. По мере необходимости их всегда можно дополнить.

Во-вторых, ряд алгоритмов из этого перечня был разработан: одни достаточно полно, другие лишь частично, но те и другие уже применялись в реальной врачебной практике. Проблема здесь в том, что созданы они давно и не развивались. С тех пор появились новые способы диагностики и лечения, а кое-где – и новые принципы. С другой стороны, многое осталось неизменным, в других случаях достаточно заменить устаревшие медикаменты на современные, не меняя логики принятия решений, в третьих – добавить новый метод диагностики и встроить в рассуждение истолкование его результатов. Этим надо воспользоваться, а то, что полностью устарело, разумеется, выбросить и заменить современным вариантом.

Не надо начинать с нуля там, где уже есть предыдущие разработки. В них надо увидеть, оценить и передать дальше сохранивший своё значение опыт предшественников, таких же хороших врачей, как те, с которыми алгоритмы развиваются сегодня. **Это принцип: алгоритмы действий врача – способ передачи ценного опыта как вири, множеству врачей, работающих сегодня, так и в будущее.**

Разработчик должен обладать природной склонностью классифицировать, «раскладывать всё по полочкам», различать детали, устанавливать их связи и субординацию. Нужен логический склад ума (в противовес художественному складу, когда сложные явления воспринимаются целиком, а не по деталям). Конечно, надо быть врачом, но вовсе не обязательно - знатоком в той профессии, для которой создаётся алгоритм. Даже лекарем быть не обязательно – достаточно иметь медицинское образование. Зато необходима вера в своё дело, осознание того, что в любой области практической медицины процесс принятия врачебных решений можно выстроить в виде строгой логической процедуры, а потом отдать её врачам в качестве программы действий. При этом надо уметь внушить свою уверенность другому участнику диалога. Необходима и большая работоспособность: разработка алгоритма – процесс трудоёмкий (он не сводится к диалогам с экспертом, а те и сами по себе требуют немалого труда). Соответственно, разработчик должен находить для этой деятельности значительные ресурсы времени.

Впрочем, оценка свойств и условий, которыми должен обладать разработчик, не имеет практического значения – тот, кто ими не обладает, просто не возьмётся за создание алгоритмов. Другое дело – второй участник диалога, специалист. Разработчик должен его выбрать, а потому полезно наметить критерии выбора.

Кто может быть экспертом. Партнёром разработчика должен быть опытный лекарь, врач, повседневно занятый лечением больных в той области, для которой создаётся алгоритм. Это должен быть человек, увлечённый своим делом, современными возможностями своей профессии, но в то же время уже понимающий, что на практике в ней совсем не редко допускаются неточности, нерациональности и даже прямые ошибки, которых теоретически следовало бы избегать, но избежать не удаётся. Лучший вариант, когда он заведует отделением, то есть и сам лечит, и несёт ответственность за действия своих ординаторов. Такой эксперт при работе над алгоритмом будет очень ответственно относиться к каждому своему слову.

Он должен быть знающим, современным специалистом, однако пробелы в его эрудиции не страшны. По ходу работы они обнаружатся и будут пополнены обращением к литературе и к его коллегам. Очень важно, чтобы этот специалист участвовал в разработке алгоритма с расчётом, что сам будет первым его пользователем, чтобы он работал на себя, на своё отделение. С этим пожеланием сопряжено ещё одно: в своём учреждении он должен пользоваться таким авторитетом, при котором ему не могли бы помешать в использовании алгоритма ни старшие, ни младшие.

Самое важное условие – способность к логическому мышлению. В этом отношении врачей можно условно разделить на три категории. У одних логическое мышление выражено очевидным образом: на вопрос они отвечают чётко, при наличии разных вариантов ответа начинают загибать пальцы, называя альтернативы, сами развивают начатую разработчиком мысль, перечисляя условия и аргументы для её различных поворотов. Для разработки алгоритма это самая ценная категория специалистов, но, к сожалению, очень редкая.

В большинстве своём врачи рассуждают не очень чётко, особенно в отрыве от реального пациента, не у его постели. Перед таким собеседником надо не только ставить вопросы, но и подсказывать первоначальные ответы, возвращаться для проверки к только что сказанному, напрямую допытываться, всё ли сказано, не упущено ли что-нибудь, не забыты ли какие-нибудь детали, ограничения, обстоятельства. Чаще всего именно с этими врачами приходится иметь дело разработчику. Но по ходу работы они, уловив стиль рассуждения, начинают выдавать информацию всё активнее, полнее и точнее.

Редко, но всё же встречается и третья категория – врачи с ярко выраженным художественным способом восприятия действительности, почти не способные к логическому её описанию. Они могут быть очень хорошими врачами и заведовать отделениями, будучи примером искусной работы для своих ординаторов, но абстрактно излагать правила поведения врача в умозрительно заданной ситуации не способны. На прямые и конкретные вопросы они обычно отвечают «так просто не скажешь», «тут много разных условий», «надо посмотреть больного», «должна подсказать интуиция», «всё зависит от обстоятельств» и т.п. Спросите, что это за обстоятельства, и такой врач ответит, что они разные и так сразу перечислить их нельзя. Свой опыт они могут передать только показом, а не рассказом. В качестве участников разработки алгоритма они не годятся.

Стоит иметь в виду, что сама узкая специальность врача может способствовать (или не способствовать) тренировке и развитию логического мышления. К чёткому рассуждению о действиях особенно склонны акушеры (в меньшей степени гинекологи), урологи и нефрологи, врачи инфарктного блока, врачи бригад скорой помощи, педиатры большинства специальностей. На другом полюсе – гастроэнтерологи, кардиологи (за исключением острых ситуаций), пульмонологи и особенно реаниматологи. Здесь разработчику надо быть особенно внимательным в выборе собеседника.

Может ли разработчик, если он сам специалист в той области, для которой создаётся алгоритм, обойтись без собеседника? Я отвечаю на такой вопрос отрицательно. Независимый собеседник – то продолжатель мысли, то её инициатор, то оппонент – нужен в этом

деле всегда. Собственные размышления могут быть самыми изощрёнными, но не способны сравниться с эффективностью живой беседы.

Общая схема взаимодействия. Приступая к работе, в пределах конкретной специальности, разработчик должен определить, к какому алгоритму и к каким классам алгоритма будет относиться разработка. При необходимости надо к имеющимся классам добавить новый или даже создать новый алгоритм со своими классами. Выбрав в качестве предмета работы алгоритм с его классами, можно идти на первую рабочую встречу со специалистом. В жизни, впрочем, чаще бывает наоборот: сначала находится заинтересовавшийся алгоритмизацией врач и разработка осуществляется для него, для его специальности.

Разработка алгоритма в диалоге со специалистом - это длинный ряд встреч, сеансов, которые следует чётко организовать. Для каждого сеанса надо отводить умеренный срок, например один час. Желательно, чтобы это было одно и то же время в определённые дни недели. Лучший вариант, когда дело поставлено так, что алгоритм нужен не любознательному разработчику, а самим медикам (главному врачу или заведующему отделением), входит в их планы, создаётся по их поручению. Важно и место встречи: лучше встречаться у разработчика, а не у врача, где его непременно будут отвлекать пациенты, медсёстры и другие врачи. Надо предусмотреть, чтобы в обсуждение не могли вмешиваться третьи лица, последние нежелательны даже в качестве слушателей (их время придёт впоследствии).

В распоряжении разработчика должны быть стопка бумаги, ручка, программа-редактор файлов-ситуаций и программный комплекс «Частные алгоритмы» с компьютерной базой знаний (с уже сделанными алгоритмами), к которой можно обратиться для общей ориентировки, для проверки уже сделанного, для поиска возможных аналогий, а иногда и для немедленных исправлений или ввода новых данных. Конечно, как правило, разработчик вводит всё наработанное за сеанс уже по его окончании, один, когда есть возможность тщательно обдумать и отшлифовать то, что набросано на бумаге за время встречи.

Врачу-специалисту очень полезно иметь программный комплекс «Частные алгоритмы», чтобы самостоятельно знакомиться с конечным продуктом разработок, но к самой встрече с разработчиком ему не надо вооружаться ничем. Рыться в медицинских руководствах и справочниках во время сеанса некогда, задача разработчика – набросать на бумаге то, что врач держит в уме. Уточнять те или иные утверждения с помощью литературы или других врачей эксперт будет потом, после сеанса, без разработчика. Это – «до-

машнее задание» для специалиста, подобно тому, как для разработчика заданием к следующей встрече является превращение набросанного за сеанс эскиза во фрагмент дерева решений. Следующий сеанс начинается с того, что участники показывают друг другу результаты «домашних заданий» и окончательно завершают предыдущую работу. После этого можно двигаться дальше.

Подготовка к сеансу работы. Специалист никак не готовится к очередному сеансу (кроме «домашнего задания»). Разработчик готовится всегда. Если предстоит первый сеанс выбранной для разработки темы, он должен представить себе и потом показать врачу место этой темы во всей системе алгоритмов: алгоритм, класс, все ситуации, в которых, по его представлениям, может возникать намеченная к разработке тема. Медицинское образование и предыдущее общение со специалистом вполне позволяют это сделать.

Далее, надо наметить начало дерева решений в той мере, в какой это может представить себе разработчик: первый вопрос и возможные ответы на него. Никакой беды в том, что это будет сделано не полно и не лучшим образом, нет. Важно заготовить канву для начала работы. Предложенная собеседнику, она поставит перед ним первые, ближайшие цели и наметит ограничения. Специалист увидит в этом серьезность намерений разработчика, его старательность и, разумеется, недостаточную осведомленность. Так ему предоставляется повод развернуться самому, показать свои знания. Так создаются условия для хорошего начала диалога.

а) Определение темы. Новую тему для очередной встречи надо согласовать со специалистом заранее. Желательно, чтобы предложение исходило от него, соответствовало его актуальным запросам. Основанием для выбора могут быть реальные проблемы лечебно-диагностического процесса в учреждении (сложные случаи, споры и ошибки врачей, новые правила, которые всегда вводятся с трудом и т.п.), частое возникновение одних и тех же ситуаций или, наоборот, редкие, исключительные ситуации, наконец, просто субъективный интерес специалиста.

Разработчик может делать и свои предложения. Тактически выгодно сначала взяться за то, что в практике врача-специалиста встречается наиболее часто. Дело в том, что когда алгоритмом охватывается более 70 процентов пациентов, проходящих через медицинское подразделение, его полезность становится очевидной для врачей и он легко вводится в эксплуатацию. Второе основание для выбора – сложность и ответственность клинической темы: здесь наилучшим образом демонстрируется эффективность алгоритмизации.

Если эксперт возражает против предложений разработчика, то с этим следует считаться, но есть два основания для возражений, которые надо решительно отвергать. Одно сводится к тому, что тема слишком проста и обычна, второе – к тому, что она слишком сложна. На деле, при разработке, оказывается, что за простотой скрываются существенные нюансы, учёт которых повышает эффективность медицинской помощи, а что до сложности, то именно алгоритмизация позволяет распутать любые хитросплетения и сделать сложное ясным, понятным, непротиворечивым.

б) Подготовка начальных эскизов. Предварительное согласование темы нужно разработчику, чтобы подготовить свои предложения о том, как алгоритм будет вводить врача в эту тему: из каких классов, в результате каких действий, после каких первых вопросов и ответов.

Может оказаться (и по мере разработки алгоритмов встречается всё чаще), что выбранная тема в той или иной мере уже затрагивалась в других местах, и тогда надо предусмотреть согласованность с ними. Некоторые другие разделы в алгоритмах могут быть созвучны намеченной теме по конструкции (формулировки вопросов и ответов, последовательность структурных элементов, особенности циклов), - тогда необходимую форму можно просто скопировать.

Если разработка темы начнётся с первой ситуации класса, надо заранее заготовить её набросок. На листе бумаге при альбомной ориентации: очерчивается верхний колонтитул. В нём записывается номер, присвоенный этой ситуации (в данном случае единица), и её суть.

Далее надо сформулировать первый вопрос и ответы на него так, чтобы, по крайней мере, один ответ означал шаг к избранной для работы теме. От этого ответа разработчик будет продвигаться дальше посредством следующих вопросов и ответов, которые, пока позволяет место, будут помещаться на этом же листе, а затем переходить на следующий лист, которому тоже присваивается номер. Переход на следующий лист делается только от ответа или от рекомендации.

От каждого из остальных ответов следует начертить стрелку к записанному ниже номеру следующей ситуации, подлежащей разработке. Для всех таких ситуаций, намеченных к разработке, надо заготовить новые листы (тоже в альбомной ориентации) с верхними колонтитулами, где записываются номер ситуации и её задача или суть (обычно это суть предыдущей ситуации плюс ответ, сделанный перед переходом из неё). Так делаются заготовки (план действий) на будущее.

Обычно первая ситуация класса начинается рекомендацией «Клинический минимум» (в классах, где рассматриваются грозные события, используется рекомендация

«Срочное обследование»). Из технических соображений имеет смысл сразу отсюда перейти в следующую ситуацию (на следующий лист). Содержание клинического минимума (то обследование, которое в медицинском учреждении обязательно для начала работы с пациентом) записывается в качестве комментария на обороте листа. Разработчик может его только наметить (особенно то специфическое, что принято в данной специальности), а полное описание оставить специалисту в виде «домашнего задания» к следующей встрече.

Подготовка к встрече, на которой будет разрабатываться не первая ситуация класса, а продолжение уже намеченных или частично выстроенных линий алгоритма, ограничивается эскизом очередной ситуации (или нескольких ситуаций). Сверх того, в этом случае разработчику надо отшлифовать всё, что было разработано на предыдущей встрече, и ввести отредактированную информацию в файлы-ситуации базы знаний. При выполнении этого «домашнего задания» могут обнаружиться пробелы и сомнительные места, которые тоже надо подготовить для обсуждения со специалистом.

в) Запас свободных номеров. Когда работа начинается с первой ситуации того или иного класса, разработчик может использовать для обозначения создаваемых ситуаций любые номера (только начальная ситуация непременно обозначается единицей). Обычно стоит придерживаться последовательности чисел натурального ряда, но можно допускать и целесообразные исключения. Дело в том, что в разных классах могут встречаться ситуации, сходные по смыслу и структуре. Их удобно обозначать во всех классах одинаково.

Если класс уже частично сделан, то разработчику надо знать, какие номера уже использованы для обозначения ситуаций, чтобы ими при предстоящей встрече со специалистом уже не пользоваться. То есть полезно иметь при себе шпаргалку с уже использованными номерами.

Сеанс разработки. Итак, если предыдущая встреча закончилась «домашними заданиями», то сеанс начинается с проверки этих заданий и окончательной отделки того, что было сделано в предыдущий раз. Если заданий не было, работа начинается сразу с заготовок разработчика - с предлагаемых им первых вопросов и ответов.

а) Владение инициативой. Подготовка разработчика к встрече с экспертом нужна, чтобы побересть время участников. Но главная её цель – сразу сфокусировать внимание на определённом предмете и потом продвигаться, только отталкиваясь от него. С помощью подготовленных материалов разработчик с самого начала берёт инициативу в свои руки. На протяжении всего сеанса эту инициативу надо сохранять, не забывая в то же время, что на самом деле, по содержанию, диалог движется ответами эксперта, а вопросы разработ-

чика лишь не позволяет отклониться от ближайшей поставленной задачи, выйти за форму алгоритма.

Предложения разработчика могут быть неполными или даже неверными, но они создают отправную точку для действий. Специалист немедленно поправляет предложенную заготовку, выдаёт при этом первые порции своих знаний, разработчик старается их улавливать – и дело пошло. Надо только следить, чтобы сохранялось представление о реальной клинической ситуации, когда именно состояние пациента обязывает врача ставить конкретные вопросы, искать на них ответы и принимать обоснованные ответами решения. Речь идёт не о правилах, общих схемах и принципах (они подразумеваются и не подвергаются сомнению), а об имитационном моделировании врачебного труда во всех его деталях, относящихся к работе с информацией. Если это упускается из виду, эксперт легко сбивается на общие описания своей специальности. Для успешного противодействия уходом в профессиональные выси можно настроиться на такую схему поведения, которая всё время оставляет инициативу за разработчиком. Попробую её изложить.

Выстраиванию рационального рассуждения предшествует кратковременный этап ознакомления с той проблематикой, которая охватывается выбранным для работы классом ситуаций. Тут разработчик должен предоставить собеседнику свободу, пока не почувствует, что уже можно оттолкнуться от сообщённых ему сведений и задать свои первые вопросы.

О чём расспрашивать специалиста-медика, чтобы составить алгоритм его действий? О действиях, разумеется, и об их обосновании. Общий вопрос подразумевается: что делать? Надо только его конкретизировать.

Ответ на общий вопрос тоже подразумевается: узнать, чтобы помочь, и – узнавши – помочь. Это основная логическая последовательность, не надо только соскальзывать с неё на ходульное «диагноз – лечение». Первая часть общего ответа сразу даёт разработчику опору для активной позиции, для вопроса «как узнать?», как добыть знания, позволяющие приступить к оказанию помощи. Специалист ответит перечислением способов обследования, и его тут же надо попросить выделить среди них те, которые позволяют продвинуться к цели, к оказанию помощи.

В любой ситуации, откуда начинается или продолжается работа врача, он, опираясь на известные к данному моменту сведения, держит в уме несколько предположений о природе состояния пациента, тех предположений, от выбора между которыми зависит способ помощи. Первая задача обследования – получить аргументы для такого выбора, сделать одну из гипотез рабочей, руководством к действию. Вот о полном перечне вероятных в данной ситуации гипотез и надо с карандашом в руке попросить специалиста. В

этот момент разработчик перехватывает инициативу: теперь его вопросы станут конкретными, а ответы эксперта будут поддаваться проверке на полноту и непротиворечивость.

Разработчик просит для каждой из названных гипотез перечислить её характерные признаки, но лишь те, которые можно добыть с помощью только что предложенного обследования. Дальше он уже сам видит признаки, которыми каждая из гипотез отличается от других, выделяет их и просит специалиста следить за его заключениями и поправлять их. Дальнейшее продвижение можно изложить на бумаге по одной из следующих схем

Простая схема начинается фразой «Рабочая гипотеза...», то есть вопросом о том, что именно выбирает врач. Готовыми ответами служат все возможные варианты, для каждого из которых в комментариях указываются отличительные признаки. Такой способ предпочтителен или даже единственно возможен, если доступные обследованию сведения не могут быть использованы в качестве бесспорных различительных признаков, если они хоть порознь, хоть в разных сочетаниях позволяют предположить лишь большую вероятность одной гипотезы в сравнении с другой. Здесь всегда есть риск неверного выбора и его надо будет предусмотреть, позаботиться о мерах дальнейшего наблюдения, оставаться в готовности пересмотреть рабочую гипотезу и изменить тактику. Здесь, как правило, надо предусмотреть и ответ «предпочтения затруднены».

Другая схема уместна в случаях, когда гипотеза основана либо на бесспорных (патномоничных) симптомах, либо тоже на таком сочетании признаков, которое можно считать специфичным, которого при других альтернативах не бывает. Тогда выстраивается последовательность вопросов об этих признаках с ответами типа «имеется / отсутствует / сомнителен» или в виде количественных оценок. Надо быть готовым к тому, что эксперт выложит не только то, что действительно позволяет сделать выбор, но и всю огромную симптоматику болезни, включая детали, не играющие дифференциально-диагностической роли. Об этой роли каждого признака надо специально допытываться и отсеивать то, что не может служить аргументом для выбора. Эту схему можно комбинировать с предыдущей.

Когда описаны все возможные в данной ситуации предположения, от которых зависит способ помощи, и изложен порядок выбора из них, задача «Узнать, чтобы помочь» решена, зафиксирована на бумаге. Теперь от каждой из возможных гипотез надо продолжить путь, отвечая на вопрос «Чем помочь и как оценивать помощь».

Здесь разработчик вообще не выпускает инициативы из рук. Он задаёт вопросы о методах помощи и слежения, о минимальных сроках, за которые надо оценить продвижение к искомому эффекту. Он записывает 4 основных результата этапа лечения и уточняет вариации "ухудшения" и особенности результата «без перемен». Он сам предлагает по-

вторить этап в случае улучшения. Он спрашивает, что надо делать при неудовлетворительных результатах этапа, а когда собеседник задерживается с ответом, сам предлагает на выбор возможные решения: повторить курс, изменить лечение, пересмотреть рабочую гипотезу, пригласить консультанта. Как только эксперт указал, что именно уместно в данном пункте, следуют очередные вопросы: как именно изменить, какую другую гипотезу рассмотреть и как для этого обследовать, как подготовиться к консультации, какие вопросы поставить перед консультантом и какое решение консультант может принять (перечень альтернатив).

И только в случае, когда обеспечен наилучший возможный результат этапа, разработчик уступает инициативу собеседнику, задавая лишь общий вопрос: что делать дальше? Впрочем, и здесь в части случаев уместно спросить о конкретных вариантах: не продолжить ли лечение для закрепления результата? не следует ли несколько дней понаблюдать, отменив лечение? всё закончить («закрыть линию»)? дать ли какое-то напутствие?

Разумеется, у эксперта всегда есть возможность опротестовать, уточнить, дополнить то, что на его глазах набрасывает разработчик, но только такая инициатива от него и требуется: в заданных рамках, по конкретному поводу, с ответственной аргументацией, благо возражений и исправлений без аргументации не бывает.

Всё, что рождается в этом взаимодействии, разработчик тут же заносит на бумагу. Внимание собеседников приковывается к записям. Разрастаясь, они всё время служат опорой, точкой отталкивания для движения в определённом направлении и в определённых границах. Можно, конечно, вводить информацию и сразу в файл-ситуацию, но это не лучший способ. Во-первых, черновые варианты по ходу сеанса будут неоднократно переписываться (а только бумага всё терпит). Во-вторых, внимание эксперта надо приковывать к привычному и простому объекту, а файл и не прост, и не привычен.

б) Начальные вопросы. В самом начале работы с классом ситуаций, когда врач должен ухватиться за что-то, чтобы начать рассуждение, используются типовые вопросы. Обычно это «Ведущий синдром:». В классах, поименованных как «Неотложные состояния» или «Острые ситуации» его аналогом служит «Ведущий синдром, создающий картину неотложности:». В классах с именем «Сопутствующие» в качестве первого вопроса служит «Поражённая система органов:». В классе «Диспетчер 03» для ССМП первый вопрос - «Что случилось?» (точнее, это второй вопрос, он следует после фразы «Скорая слушает. Куда ехать?»). Такие вопросы влекут за собой целую россыпь ответов. Сколько бы их ни было, желательно расположить их на одном листе. Если это не удаётся, надо один из ответов сформулировать как «прочие (см. на обороте)» и на обратной стороне листа записать все остальные ответы. Каждый ответ связывается стрелкой с номером бу-

дущей ситуации. Для этих будущих ситуаций заготавливаются листы с номером и краткой формулировкой сути в верхнем колонтитуле.

в) Наброски типовых конструкций. Описанные ранее типовые логические конструкции – не произвольная выдумка. Они отражают наличие устойчивых стереотипов поведения, выработанных врачебным опытом, диктуемых наукой, а также логикой здравого смысла. Информацию, которую выдаёт эксперт, надо на его глазах укладывать в эти конструкции, тут же заботясь об их формальной полноте.

Приступая к обсуждению каждого этапа врачебной работы, надо представить его, как элементарную клиническую ситуацию: со своей задачей, необходимыми действиями, вопросами и ответами. Сформулировав очередной вопрос надо сразу позаботиться о полном множестве ответов, в том числе и о неопределённом ответе («неизвестно», «сомнительно», «предпочтения затруднены»). Если ответ ведёт одновременно к двум разным действиям, надо сразу написать под ним «параллельно» и начертить вертикальные параллели. При допустимости множественного выбора надо сделать пометку, что здесь требуется цикл «Ещё». Если краткая формулировка ответа не является исчерпывающей, определённой и недвусмысленной, то такой ответ надо снабдить пометкой «см. на обороте» и все аргументы для его выбора, тщательно расспросив эксперта, написать на оборотной стороне листа.

Раздел, относящийся к рекомендациям, надо обозначить одной фразой с пометкой «см. на обороте», а все подробности, если они есть, записать на оборотной стороне листа. Если эти действия должны занять некоторое время и принести новую информацию, то следующий за рекомендацией вопрос должен начинаться словами «В пределах [установленного срока]...».

Если всё это делать именно так при каждом новом шаге в диалоге, то на бумаге появляется хороший черновик для «домашнего задания» разработчику, а эксперт не только видит, что его знания бережно раскладываются «по полочкам», но постепенно тоже проникается правилами такого распределения. Кроме того, он получает «домашнее задание» в тех случаях, когда не может сразу дать исчерпывающее описание необходимых действий (врачебных назначений) или точные комментарии к ответам.

Из всего, что перечислено, вызвать у эксперта затруднения может только необходимость определить очередные действия при неопределённом ответе («неизвестно», «предпочтения затруднены»). Тут разработчик должен подсказать ему основные направления: дополнительные исследования с целью снять неопределённость, ту же тактику, что и при одном из определённых ответов, параллели для того и другого, наконец, какой-то

особый путь. Такая подсказка сохраняет инициативу за разработчиком, а эксперту даёт понятные ориентиры.

г) Выбор на основе количественных, полуколичественных и качественных критериев. При выборе на основе количественных критериев, нельзя использовать пары «больше А» и «меньше А» или «больше А» и «меньше В». Должно быть три ответа: меньше А, в пределах от А до В, больше В. Если всё же используются только два ответа, то правильная формулировка – это «больше А» и «не больше А». Кроме того, ответом на вопрос о количественных показателях может быть и «невозможно определить». Это бывает редко, если касается общепринятых измерений, анализов или данных анамнеза, но всё же бывает (например, давность болезни, ещё не полученный анализ или невозможность анализа в данном учреждении). В ситуациях, когда такие случаи не выглядят невероятными, их стоит предусмотреть, предложив врачу рациональные действия и при подобных ограничениях.

Примерами полуколичественных оценок могут служить термины «высокий – нормальный – низкий», «тяжёлое – средней тяжести – удовлетворительное», обозначения стадии или степени патологического состояния. Там, где возможно, лучше заменить такие ответы на количественные характеристики. В остальных случаях к ответам надо давать комментарии с чёткими определениями: что относится к 1-й, 2-й и 3-й степени болезни, к средней тяжести состояния больного и т.д.

Такие комментарии совершенно необходимы к ответам, которые выбираются на основе качественных критериев. Примеры – диагностика этиологии поноса у ребёнка по виду кала, оценка вида сыпи при инфекционном заболевании, характеристика кашля и одышки. Здесь важно, чтобы были перечислены все те варианты, которые имеют диагностическое значение.

д) Комментарии к ответам. Эти комментарии используются для получения двух сопряжённых эффектов. Поскольку они в протокол не выводятся (но при наведении курсора на ответ открываются на экране), сам ответ можно ограничить лаконичной формулировкой, достаточной для протокола, а обоснование выбора этого ответа, все аргументы, какими бы обширными они ни были, перенести в комментарии. При необходимости здесь можно поместить формулы необходимых расчётов, целые инструкции, напоминание о том, что последует за выбором этого ответа и даже ссылки на литературу. Если логику рассуждения называют деревом решений, то комментарии к ответам можно сравнить с его богатой листвой. Эта листва позволяет предложить взгляду врача именно те медицинские знания, которые нужны ему для правильного решения в данный момент. Эти знания надо выпрашивать у эксперта и записывать со всей тщательностью. Немаловажно, что макси-

мальное использование комментариев к ответам лишний раз демонстрирует эксперту глубину проникновения алгоритма в его специальность.

е) Рекомендации в лечебных ситуациях. С самого начала важно обратить внимание эксперта на то, что лечение можно представить как процесс, состоящий из таких минимальных по времени этапов, за которые накапливается достаточно сведений, чтобы решить, достигается ли желаемый эффект и не надо ли внести в лечение поправки. Ключевой элемент здесь – срок. Он обратно пропорционален скорости и опасности патологического процесса, так что подведение этапных результатов может требоваться в одних обстоятельствах каждые 5-10 минут (в неотложных состояниях), в других – ежедневно или еженедельно, в-третьих, - раз в месяц (например, при лечении туберкулёза) и даже реже (амбулаторное наблюдение при достигнутой ремиссии). Как ни странно, врачи часто затрудняются назвать этот срок. Для начала разработчик может предложить его сам, руководствуясь тем, что чаще всего этот срок составляет четверть от всей предполагаемой длительности успешного лечения.

Каждому такому этапу соответствует лечебная ситуация, которая начинается с рекомендации о врачебных назначениях. Рекомендации эти – такая же листва на дереве решений, как комментарии к ответам. Разница в том, что они, представляют собою назначения врача и полностью выводятся в протокол. Они должны быть изложены именно как назначения: точно, кратко, в естественной для врача последовательности: диета и режим, разделы лечения, контроль. Если медикаментозное лечение допускает различные варианты, их надо указать. Из множества препаратов одной фармакологической группы надо выбрать один в качестве типопредставителя (то есть на деле он может быть заменен тем или иным аналогом). Надо записать дозы, способы введения и кратность приёма. В разделе «Контроль» перечисляются объекты, способы и ритм наблюдения (симптомы, за динамикой которых надо следить, измерения и подсчёты, лабораторные и инструментальные исследования, их частота). Тем самым определяются не только диагностические назначения врача на время лечения, но и содержание и частота его дневниковых записей.

ж) Формулировки для оценки результатов лечения и наблюдения. С экспертом надо уговориться, что оценка результатов каждого лечебного курса должна приводить к одному из четырёх выводов, присущих «лечебной ситуации». Общие формулировки этих выводов: полное исчезновение симптомов, существенное улучшение, без существенных перемен и ухудшение. Они имеют принципиальное значение для принятия решений и должны пониматься экспертом (а потом – врачами) однозначно. Поэтому их надо пояснить.

Под полным исчезновением понимается ликвидация симптомов острого заболевания или симптомов обострения хронического процесса. В выводе об улучшении принципиально важен эпитет «существенное», то есть такое улучшение, которое можно расценивать как явное, ощутимое продвижение к выздоровлению или к ремиссии. «Без перемен» - это либо сохранение симптомов и их выраженности, либо исчезновение одних симптомов болезни в сочетании с появлением других. Термин «ухудшение» достаточно выразителен, но в ряде случаев надо предусматривать особый вариант ухудшения, вынуждающий к срочному и радикальному изменению тактики (терминальное состояние, непереносимость препаратов). Для этих случаев следует использовать самостоятельный вывод под номером 5.

Формализация результатов лечебной ситуации, ничего не меняя в традиционных клинических понятиях, существенно их уточняет. Это позволяет сразу намечать дальнейшую тактику: 1 - завершение или ослабление лечения, 2 – продолжение того же лечения, 3 - продолжение или усиление лечения, 4 - пересмотр лечения, а то и диагноза, 5- срочные специфические мероприятия. Всё это, будучи объяснено, легко берётся на вооружение сначала экспертом, а потом и лечащими врачами.

з) Наброски комбинированных конструкций во время сеанса. Помимо простой последовательности вопросов и ответов, которые легко заносятся на лист бумаги (в «цифровую» ситуацию) в виде дерева решений, сведения, излагаемые экспертом, требуют использования более сложных конструкций. Эскизы этих конструкций надо делать тут же, на глазах у собеседника.

С этой целью каждый раз, когда наступает момент назначить для последующего принятия решений специальное диагностическое исследование или лечебный курс, надо зафиксировать это исследование или этот курс на отдельном листе: записать наименование, перечислить и пронумеровать все возможные результаты и обозначить этот лист не

Сит. Л. Лечение острой пневмонии.



Оборотная сторона.

Стол 15. Режим постельный. Лечебная гимнастика.

Антибактериальная терапия (препараты, дозы, способы введения, частота).

Патогенетическая терапия (препараты, дозы, способы введения, частота).

Симптоматическая терапия (препараты, дозы, способы введения, частота).

КОНТРОЛЬ. Самочувствие, общее состояние. Кашель, мокрота. Аускультация лёгких. Термометрия утром и вечером.

Обзорная рентгенограмма лёгких, общие анализы крови и мочи – всё раз в 7 дней.

Рис.3. набросок лечебной ситуации.

номером, а буквой, то есть сделать эскиз «буквенной ситуации». В лечебной ситуации на обратной стороне листа записываются рекомендации о лечении и контроле, в диагностической – при необходимости – противопоказания и способы подготовки к исследованию. Пример такого наброска представлен на рисунке 3 (это схема, а не реальный фрагмент алгоритма; как минимум, в ней недостаёт логики оценки симптомов или комментариев к ответам).

Ситуация 24. Лечение.

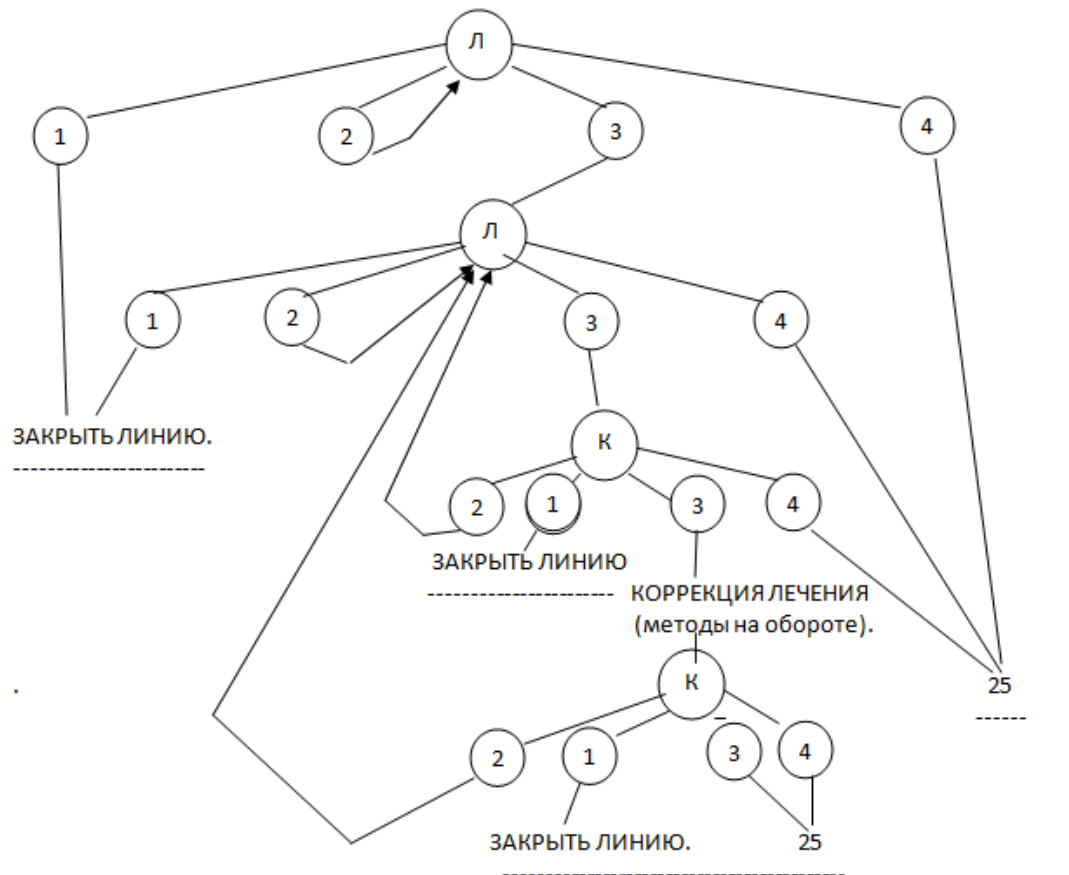


Рис.4. Набросок комбинированной конструкции. Объяснения в тексте.

Теперь, пользуясь этим эскизом, можно на другом листе быстро набрасывать те комбинации, которые следуют из дальнейших ответов эксперта на вопросы разработчика: что делать при одном результате, при другом, третьем и четвёртом. По ходу такого расспроса появляется структура, подобная представленной на рисунке 4.

Там в гипотетической ситуации 24 буквенные ситуации образуют сложные комбинации, которые, тем не менее, легко читаются благодаря стереотипному значению букв и цифр. Так, если ситуация Л завершается полным исчезновением симптомов (24-Л1), работа заканчивается. В случае улучшения (24-Л2-) следует повторить тот же недельный курс (24Л2-). При ухудшении (24-Л4-) алгоритм уводит в ситуацию 25 (24-Л4-25), где будут выясняться причина ухудшения и способы радикальных изменений в лечении (обычно это конструкции «Уточнение диагноза» и «Коррекция лечения»).

Сложнее развиваются события при выходе 3 («без перемен»). Тут лечение Л продолжается ещё неделю (24-Л3-Л-). Если и вторая неделя не приносит желательного сдвига, то на следующей неделе используют усиленный курс лечения К (24-Л3-Л3-К-). Если и этот курс не приносит изменений, то предлагается его повторить, но с включением дополнительных методов, перечисленных в блоке «КОРРЕКЦИЯ ЛЕЧЕНИЯ». (24-Л3--Л3- К3-

коррекция -К-). Наконец, если и курс К не приводит к переменам в течение 2 недель, результат 3 приравнивается к результату 4 (ухудшение) и переводит в ситуацию 25 (24-ЛЗ-ЛЗ-КЗ-КЗ-25).

Обсуждая все повороты событий в ведении пациента, такую схему надо всё время держать перед собой и экспертом, приучать эксперта ею пользоваться, приковывать его (и своё) внимание именно к тому пункту, в который привёл ход изложенных в разговоре событий. Схему можно тут же, на ходу поправлять, дополнять, переписывать, пока логика поведения врача не предстанет во всей полноте и ясности. Эксперт от такого наброска получает профессиональное удовлетворение, а разработчик – проработанный материал для выполнения «домашнего задания».

и) Консультации специалистов. Положение, когда для принятия решения специалисту не хватает своих знаний или умений и поэтому требуется консультант, отображается на листе бумаги сначала рекомендацией вида «Консультация кардиолога [хирурга, эндокринолога и т.д.]». В части случаев здесь надо указать, как следует подготовиться к консультации. Затем следует фраза «Заключение консультанта: » и отходящие от неё все возможные заключения консультанта, существенные для принятия дальнейших решений, включая и неопределённое («неизвестно», «сомнительно», «не удаётся установить»). Всегда следует поинтересоваться логикой консультанта. Если её можно изложить в виде вопросов и ответов, ведущих к выбору из перечня альтернатив, то так и надо сделать. В других случаях можно воспользоваться комментариями к каждой из альтернатив, с перечислением аргументов, позволяющих предпочесть именно её.

к) Разрешение сомнений. Исходная позиция разработчика состоит в том, что он принимает все, что утверждает специалист, на веру. Однако по мере углубления в тему, а также в силу своего медицинского образования он временами обнаруживает, что не со всем может согласиться. Эти сомнения надо немедленно высказывать. Чаще всего специалист даёт разъяснения, которые он считал с высоты своих знаний излишними. Это всегда полезная информация, её надо использовать в алгоритме, чтобы там не было сомнительных или двусмысленных мест ни для специалистов, ни для дилетантов.

В других случаях эксперт признаёт и исправляет свою ошибку, неточность, недоработку. Чаще всего такие недоработки заключаются в слишком общих положениях там, где для алгоритма нужна полная детализация, учёт всех существенных для принятия решений нюансов.

Разработчик может обнаружить, что один участок алгоритма необъяснимо противоречит другому. Как правило, обсуждение такого противоречия приводит к уточнению

обоих участков. Дело в том, что эксперт не сразу усваивает, что разработчик не лекцию слушает, а расписывает поведение врача, что ему нужны не обобщения, а детали, не просто правила, а правила со всеми исключениями.

Наконец, бывает и так, что сомнения разработчика ставят специалиста в тупик, он соглашается с правомерностью вопроса, но не готов на него ответить. Тогда поиск решения включается в его «домашнее задание» (размышления, обращение к литературе и к другим специалистам).

л) Заключительная проверка сделанной работы. В конце сеанса разработчик должен просмотреть все только что сделанные записи, убедиться в их читабельности и понятности, пометить те места, которые надо будет пополнить, когда специалист выполнит своё «домашнее задание».

Теперь остаётся поиграть в алгоритм: разработчик – за лекаря, специалист – за природу, за болезнь, за житейские обстоятельства. Разработчик с набросками в руках задаёт вопрос из алгоритма, получает от специалиста ответ, задаёт следующие вопросы и получает ответы, наконец, приходит к рекомендации и сообщает специалисту о «своих» решениях. Иными словами, проигрываются элементарные клинические ситуации. Всё хорошо, пока специалист выражает одобрение разумным вопросам и верным решениям разработчика. Но может статься, что он в условиях, когда внимание действительно сосредоточено на воображаемом, но конкретном больном, обнаружит пробелы или неправильности. Тогда дефектные места в алгоритме придётся дополнять или переделывать.

Наконец, специалист выражает удовлетворение, а то и удивление тем, как грамотно действует дилетант-разработчик там, где не всегда сразу сориентируется и профессионал. В этот момент надо подчеркнуть, что именно он, специалист, пропитал записи разработчика своими знаниями и логикой и что теперь они станут достоянием многих врачей. На этом, прихватив «домашние задания» (у разработчика – листки с набросками, у эксперта – памятка о том, что надо уточнить с помощью литературы и коллег) и наметив дату и час следующей встречи, можно разойтись.

После сеанса. То, что накопилось в набросках за время сеанса (а записывать надо было все детали, ни в чём не полагаясь на память), теперь надо перенести в файлы-ситуации. Перенести не откладывая, иначе не избежать досадных упущений, которые потом непременно обнаружатся и будут выглядеть как невниманье к эксперту. Нельзя откладывать и потому, что к следующему сеансу надо демонстративно обработать всё, что накоплено в предыдущем. Работы предстоит много, творческой и технической, времени всегда мало, отсюда важнейшее требование – работать методично.

Сначала необходимо создать с помощью программы-редактора файл-ситуацию, с которого начинается разработанная с экспертом тема, и заполнять его принесенным со встречи содержанием, определяя, с каких мест следует переходить в другие файлы. Этим файлам надо тут же присваивать числовые или буквенные коды и сразу заготавливать их для последующего заполнения (редактор не позволит случайно использовать уже задействованные номера).

В строке 000 каждого заготовленного файла надо сразу указывать его задачу в поле text. Если для этого понадобится пространное описание, его можно продолжить в поле comm. Редактор автоматически занесёт эти заготовки в специальный архив, отражающий работу за сегодняшний день, так что забыть о них будет нельзя. Не беда, что в ходе работы окажется, что одни заготовки лучше объединить в общем файле, а другие разукрупнить. Это всегда легко сделать, а сейчас, с самого начала, надо создать основу, охватывающую обсуждавшийся сегодня предмет и указывающую, куда надо помещать все детали.

Нельзя держать общий план в уме, не надо его сколько-нибудь долго обдумывать — надо делать заготовки и приступать к детализации, не отрываясь от клавиатуры. Думать, обдумывать разные варианты надо не до, а во время этой работы. Такое правило основано на том, что общий план волей-неволей уже был создан в уме во время встречи с экспертом, а шлифовка осуществляется лучше всего только тогда, когда видишь разные варианты, благо на экране они выразительны и их легко перебирать.

После того, как таким образом поле деятельности очерчено, надо сначала сделать буквенные ситуации, а если они существовали раньше, то проверить, в полной ли мере их содержание соответствует применению на новом месте. Если оказывается, что для использования уже существующей «буквенной» ситуации достаточно применить более совершенные формулировки вопросов, ответов или рекомендаций, это можно сделать, но очень осторожно. При необходимости более радикальной подгонки к специфике новой задачи (новые ответы, дополнительные вопросы), лучше сделать отдельную «буквенную» ситуацию для случаев, подобных этой новой задаче, а если других подобных случаев не предвидится, то и вообще отказаться от «буквы» в пользу цифровой ситуации.

Когда буквенные ситуации созданы, заполнение цифровых ситуаций превращается в последовательное, не прерывающееся изложение вопросов, ответов и рекомендаций, связи которых видны на экране со всей выразительностью. Когда та или иная линия рассуждения не может быть продолжена, потому что необходимая для этого информация от специалиста ещё не получена, эту ветвь надо закончить заглушкой: в поле Kz отметить переход на файл v9999999. Такой приём, во-первых, позволяет потом легко найти точки дальнейшего развития, а, во-вторых, защищает пользовательскую программу от аварийно-

го завершения: файл-ситуация с таким именем существует и обеспечивает продолжение работы.

Имея перед глазами заполненный новый файл-ситуацию, легко проверить перечень ответов к вопросу на полноту, а сами ответы и комментарии к ним – на понятность и недвусмысленность. Читая вопросы и ответы, нетрудно представить, как они будут образовывать фразы в протоколе рассуждений. Форматируя рекомендации о лечении и наблюдении за больным, легко обнаружить те или иные пробелы.

Любая нелогичность, недосказанность, любое несоответствие между структурными элементами файла здесь бросаются в глаза. Если устранить их на основе полученных от эксперта сведений не удаётся, надо сделать себе пометки (не в уме, а на бумаге) для следующей встречи со специалистом.

Теперь надо лишний раз проверить, всё ли взятое от эксперта, зафиксированное в записях и набросках пущено в дело. Важны все мелочи, вся фактура, все повороты мысли. Наконец, сделанное надо проверить в работе: запустить пользовательскую программу, демонстрационный режим, пройти все новые линии, каждый раз доводя до формирования протокола «Ход рассуждения» и записывая этот протокол в текстовый файл (программа это предлагает). На экране надо внимательно рассмотреть все комментарии к ответам, остальное лучше всего оценить по протоколам. Когда обнаружится необходимость редакционных поправок, их непременно надо тут же внести и посмотреть, как это работает, заново.

Необходимые доработки, если они обнаружились, станут предметом следующего сеанса работы с экспертом. Если они незначительны или вообще отсутствуют, то программный комплекс «Частные алгоритмы» надо обновить и сообщить об этом эксперту, чтобы он мог взять последнюю версию и тоже посмотреть получившийся результат.

Сверхзадача разработчика. Эксперт сотрудничает с разработчиком из любопытства, только по доброй воле. В лучшем случае она подкреплена просьбой его руководителя (начмеда, заведующего отделением, профессора клиники). Станет скучно – встречи прекратятся. Поэтому разработчик должен постоянно заботиться о том, чтобы встречи проходили интересно, каждый раз давали интересный и явно полезный результат. Сверхзадача этих встреч – сделать из эксперта сторонника алгоритмизации врачебной деятельности, пусть не настолько увлечённого, чтобы всецело ей отдаваться, но уже понимающего, что это такое и как это может способствовать качеству медицинской помощи. Попробую изложить своё представление о том, как продвигаться к этой цели.

Прежде всего, с экспертом надо говорить только о его повседневной деятельности – никаких общих рассуждений о врачах, об их ошибках и о благе, которое несут алгоритмы. Всё внимание – конкретной работе специалиста с его больными. Здесь он знает всё, здесь он всё умеет, здесь разработчик – его любопытствующий гость. В этой роли гостя в интересном доме, у хозяина, который охотно всё показывает и объясняет, разработчик должен пребывать постоянно.

Однако разработчик – гость активный и умеющий взглянуть на то, что ему показывают, с неожиданной для хозяина стороны. Хозяин слегка удивляется, гость этому радуется, а в какой-то момент интерпретации гостя образуют вместе некую стройную композицию, которая удивляет и радует обоих. Гость чуть больше осваивается в доме, хозяин волей-неволей начинает поглядывать на собственные экспонаты его глазами. Увидеть своё собственное в ином свете, неожиданно узнать о нём больше, понять его глубже – это всегда интересно. И чем дальше, тем больше. При этом гость не упускает случая подчеркнуть, что дело не в нём, а в самом предмете и что всё интересное сказано самим владельцем, а он, гость, лишь кое о чём порасспросил. Алгоритм создаётся из знаний эксперта, разработчик – только умелый составитель.

Созданию такого умонастроения должны служить все атрибуты встреч: упомянутые эскизы клинических ситуаций на глазах у эксперта, придирчивость к аргументам, скрупулёзное внимание к деталям комментариев, проверочная игра во врача и судьбу в конце сеанса, завтрашний показ в пользовательской программе того, что набрасывалось сегодня, со всеми деталями и с протоколами рассуждений.

Вместе с экспертом надо радоваться неожиданностям, когда обнаруживается, что в якобы простых ситуациях есть свои каверзы и ловушки, которые теперь видны, или когда сложная и трудная тактика расписывается с такой точностью, при которой ошибки становятся невозможными. Радуюсь, надо ненавязчиво замечать, что без алгоритмов врач в эти ловушки попадает, эти ошибки делает. А с алгоритмами уже не попадёт, не сделает.

Непременным условием вовлечения эксперта в новую систему представлений о его собственной работе является энергичный темп взаимодействия. Паузы между встречами надо сокращать до минимума, встречи не должны срываться по вине разработчика. Продуктивность каждой встречи должна быть обеспечена полной подготовкой к ней разработчика – это, в свою очередь, побуждает хорошо готовиться и эксперта.

Наконец, надо зафиксировать участие эксперта в разработке соответствующего класса ситуаций – в первой ситуации этого класса записать его фамилию в комментариях к строке 000. В главном меню пользовательской программы есть функция «О системе и

участниках её разработки», там выводятся сообщения обо всех, кто вложил свой опыт и труд в создание алгоритмов.

Часть 4. РАЗВИТИЕ АЛГОРИТМА В ХОДЕ ЭКСПЛУАТАЦИИ.

Алгоритм можно считать работоспособным только по факту: тогда, когда им стали пользоваться врачи. Однако его никогда нельзя считать завершённым. Всегда есть вероятность, что появится такая ситуация, которую составители не учли. Всегда возможно, что кто-то пытливый на примере своего пациента предложит разумные дополнения. Наконец, лечебное дело развивается, появляются новые знания, методы и средства, - всё это алгоритмы должны впитывать. Их структура позволяет быстро и без особого труда вносить большинство необходимых изменений. Значительные усилия требуются лишь в те периоды, когда в той или иной специальности появляются новые направления, особенно если ими вытесняются старые.

До ввода алгоритма в эксплуатацию надо соблюдать правило, согласно которому трудятся только двое – разработчик и эксперт. Но перед самым вводом полезно показать сделанное ещё одному опытному врачу. Он обязательно выдаст не относящиеся к делу общие сомнения и опасения (это надо стерпеть), но свежим взглядом вполне может обнаружить упущения и спорные моменты, не замеченные ранее именно потому, что они были на виду. Это и требуется: незамысленный взгляд. То, что на поверхности, надо обнаружить и устранить заранее.

С момента ввода алгоритма в эксплуатацию вопросы и замечания последуют от лечащих врачей. Тут с самого начала приходится вносить ясность в вопрос об ответственности. Он решается провозглашением двух правил. Первое: следуйте указаниям алгоритма. Второе: отклоняйтесь от алгоритма, если не согласны с его указаниями, не понимаете их или не можете их выполнить; однако об отклонениях известите заведующего отделением. То есть ответственность врача за его действия остаётся такой, как была, а о возникших коллизиях и связанных с ними предложениях оповещается руководитель. От него и получает разработчик сведения о проблемах.

Систему оповещения разработчика и взаимодействия с ним надо продумать. Она должна предусматривать передачу тех сведений, которые позволяют ему сориентироваться заранее. Лучше всего это обеспечивает файл с протоколом «Ход рассуждения» и пометки к нему. Далее, разработчик должен иметь возможность обсудить проблему с заведующим, с лечащим врачом, а потом и с экспертом. Реагировать на проблемы, учитывая, что речь идёт о непрерывном производственном процессе, надо как можно быстрее.

В идеале уже через день-два исправленный вариант может и должен появиться на компьютерах врачей.

Часть вопросов и недоумений бывает вызвана только фактором новизны. Тут достаточно простых разъяснений. Когда алгоритм становится привычным, такие вопросы исчезают. Другое дело – обоснованные реальной практикой замечания о неполноте, нелогичности и других недостатках содержания. Если исправлять эти недостатки немедленно, то на первых порах их выявление даже усилится: врачи видят, что к ним прислушиваются, и понимают, что своей активностью пополняют общую копилку. Среди них встретятся такие, которые выдвинут дельные предложения. То же постарается делать руководитель подразделения, если увидит (если ему показать) в алгоритмах эффективный способ воздействия на качество работы врачей. Наоборот, если своевременной реакции разработчика и эксперта не будет, интерес практиков к совершенствованию алгоритма очень скоро иссякнет, они просто станут его обходить. Понятно, что доводить до этого нельзя.

У предложений, поступающих от лечащих врачей, особая ценность: они порождены реальными событиями, так что поправки можно вносить и проверять по свежим следам, с историей болезни в руках (замечания, не привязанные к истории болезни, надо отклонять). Поэтому первое, что необходимо сделать, – это спросить врача, как надо, как он поступил с этим больным на деле, и понять, что именно его озаботило: несоответствие алгоритма тому, что привычно, или тому, что должно. Полученную информацию надо обсудить с другим врачом (лучше – с заведующим отделением), стараясь уточнить, что в обнаруженной проблеме типично, что можно счесть сугубо индивидуальной спецификой и не включать в алгоритм, какие ещё несоответствия могут возникать в этом месте алгоритма. После этого пора всё показать эксперту, с которым составлялся этот раздел алгоритма. Если он в обозримый срок недосыгаем, надо прибегнуть к помощи другого опытного специалиста.

Всё упрощается, если эксперт работает в том учреждении, где внедрён алгоритм. Тогда вопросы поступают к нему, и уже он предлагает разработчику внести изменения. Но с кем бы ни велось заключительное обсуждение, разработчик должен «играть» на стороне первоначального варианта, стараясь таким приёмом проверить новые предложения на прочность.

Когда исправления сделаны и у пользователей установлена обновлённая версия программы, остаётся спросить у инициатора изменений, доволен ли он, чтобы убедиться, что не осталось никаких недоразумений и лишний раз настроить врачей на активное отношение к недостаткам их необычного инструмента.

Так как при совершенствовании алгоритма в ходе его эксплуатации перестаёт действовать правило «третий – лишний», новые предложения могут вызывать споры. Возникшие с подачи лечащих врачей, они лучше всего разрешаются при очной встрече спорящих сторон в присутствии разработчика (он, как правило, «играет» за старый вариант). Иной раз требуется обращение к литературе или к авторитетному третьейскому судье.

Отдельно надо сказать о случаях, когда спорщики не уступают друг другу, когда веские аргументы есть и у одной, и у другой стороны (обычно это касается методов обследования, видов и сроков лечения, способов и частоты контроля). Тут от разработчика требуется специальное внимание. Чаще всего подобный оборот дела – мнимый тупик. Он может указывать, во-первых, на допустимость как одного, так и другого варианта поведения врача, а, во-вторых, на недостаточную детализацию ситуации в алгоритме, так что стоит задать вопрос, разделяющий её на два варианта событий, и окажется, что уже существующий фрагмент точно соответствует одному из них, а новые предложения – другому.

Если удаётся подобная детализация, то надо к общему удовлетворению сделать такую переделку. Что же касается допустимости обоих вариантов, то тут есть два способа коррекции алгоритма. Когда предмет спора – детали в рекомендациях, нередко удаётся разрешить его включением обоих предложений, используя союз «или», допуская тем самым не регистрируемый в протоколе выбора врача. Если же речь идёт о радикальном различии, например, о разных системах лечения или разной тактике обследования, то в алгоритме надо задать явный вопрос о том, какой из равноценных путей врач выбирает, а потом проложить эти равноценные пути. К ответам на этот явный вопрос можно дать комментарии о достоинствах, недостатках, условиях предпочтительного использования каждого образа действий (доступность, эффективность, вероятность осложнений, ограничения в связи с возрастом и сопутствующими заболеваниями, стоимость и проч.). Осуществив такие усовершенствования, разработчик обогатит участок алгоритма знаниями о более дифференцированной, более индивидуализированной тактике, оснастит суждениями, к которым он подвёл врачей своими вопросами.

Сколько бы врачей в ходе эксплуатации алгоритма ни привлекалось к его совершенствованию, неизменно должно соблюдаться правило не реализовывать новшества без согласия того эксперта, который участвовал в первоначальной разработке. Эксперты должны быть уверены, что сделанное ими не будет ни выброшено, ни искажено, что полученная от них информация оберегается как важнейшая ценность. Если эксперт сменился, то разработчик должен сам отстаивать сделанное ранее до тех пор, пока его новый партнёр не представит убедительных доказательств в пользу перемен.

Помимо выявления дефектов в разработанных ситуациях алгоритма, его реальная эксплуатация в лечебном учреждении даёт разработчику ещё и другую важную информацию: она позволяет учесть, как часто врачи выходят на неработанные линии, на заглушки, в каких местах. Те места, где это происходит чаще, и надо дорабатывать в первую очередь, с ними и надо идти к эксперту.

Чем интенсивнее и шире используется алгоритм действий врача, чем методичнее реагирует разработчик на все вопросы и замечания врачей, тем быстрее выявляются и устраняются все шероховатости. Но с течением времени появляются новые средства и способы, новые воззрения, новый опыт. То тут, то там возникает необходимость включить это новое в алгоритм, благо тогда этими новшествами воспользуются к месту, по показаниям и те, кто о них узнал, и те, кто про них не слышал. Стало быть, если алгоритм используется, то разработчик и эксперт будут нужны всегда.

Часть 5. ЧЕГО НЕ СЛЕДУЕТ ДЕЛАТЬ РАЗРАБОТЧИКУ.

Допустим, что идея алгоритмизации действий врача будет жить, что алгоритмы будут использоваться. Тогда стоит заранее поразмыслить о возможных ошибках тех активных последователей, которые захотят их развивать, совершенствовать, переделывать. А желание переделывать непременно возникнет у каждого вникнувшего в то, что уже есть, у каждого новообращённого.

Оговорюсь, что для подобных размышлений у меня есть только самонаблюдение и предубеждения. Последние сводятся к тому, что увлечённые неофиты (другие в это дело не ввяжутся) всегда хотят плыть своим путем, не глядя на карту, а тем паче в лоцию. Они считают, что их «продвинутость» сама по себе оберегает от подводных камней. Понимаю, что предубеждения граничат с брюзжанием или уже брюзжание, но тема ошибок и опасностей, как уже сказано, близка мне смолоду, так что не буду изменять себе, несмотря на отсутствие достаточного фактического материала и неизбежность повторений того, что уже было затронуто под другими углами зрения.

Не создавайте без нужды новых сущностей («брита Оккама»). Архитектура системы алгоритмов действий врача продумана исчерпывающим образом. Значения терминов «алгоритм», «приложения к алгоритмам», «класс» соотносятся с медицинскими специальностями, особенностями места работы, представлениями врачей о синдромах. Понятия «файл-ситуация», «цифровые», «буквенные», «лечебные» и «специальные диагностические» ситуации отражают объективно существующую этапность действий лечащего

врача и своеобразие этапов. Структура файла-ситуации (задача, вопрос, ответы, рекомендации, срок рекомендации, комментарии) точно соответствует языку лечащего врача, структуре его рассуждения. Вся эта анатомия системы алгоритмов, связи между её членами, их соподчинённость, - всё соответствует природе лечебно-диагностического процесса. Опытом разработки и эксплуатации алгоритмов доказано, что это то, что необходимо и достаточно. Здесь нечего совершенствовать. Эти термины надо применять только в соответствии с их определениями. Эти понятия и только их надо использовать в развитии и совершенствовании алгоритмов.

Соблазн что-то переименовать, использовать термин не по назначению, «осовременить» язык жаргонными выражениями, сокращениями слов, ненужными англицизмами и проч. может возникнуть, особенно на первых порах, – его надо подавлять. Вся система алгоритмизации врачебной деятельности построена на определённости понятий, на точности терминов, на чистоте и естественности языка.

Не надейтесь на память. В основе деятельности разработчика лежит собирательство. Он подбирает крупицы врачебного опыта везде, где они ему попадают: в сеансах работы с экспертом, при объяснениях с лечащими врачами, при участии в спорах. Суждения, которые он при этом слышит, которые имеют прямое отношение к обсуждаемой теме, надо записывать. Но не следует проходить и мимо тех интересных замечаний, которые могут относиться к предметам, уже обсуждённым или намеченным к обсуждению. Особого внимания заслуживают те случайные ремарки эксперта, от которых он сам сразу отказывается, считая, что это не к месту, не имеет значения, редко бывает и проч.

Многие из случайных и отрывочных суждений могут представлять ценность в качестве отправных точек для дальнейшего обогащения алгоритма. Надо стараться их распознавать, подхватывать, уточнять. И, конечно, записывать, чтобы потом к ним вернуться. Ни на свою память, ни на то, что эти суждения будут высказаны в другое, более удобное время, полагаться нельзя.

Но и все записи, все наброски и пометки – не гарантия, что мысли не пропадут. Сделанные по ходу беседы, они – только знаки, которые должны вызывать ассоциации, воспоминания о том, что сверкнуло в уме и обещало развитие. Пройдёт немного времени, и знаки потускнеют, ассоциации пропадут. Поэтому с записями надо поработать в тот же день, не откладывая: определить их место в алгоритме, выбрать или создать файлы-ситуации и приступить к вводу вопросов, ответов, рекомендаций и комментариев.

База знаний, программа-редактор и пользовательская программа устроены так, что разработчик может делать наброски и оставлять их для последующего развития, не опасаясь

ясь, что они повредят пользователю, заботясь только о том, чтобы не потерять крупницы полученных сегодня знаний. Ввод текстов в специально организованную структуру основательно дисциплинирует мысль, способствует её целенаправленному развитию. По свежим следам это позволяет хорошо продвинуться. Упустишь время - следы сотрутся.

Не делайте ничего, что может задеть эксперта. Опытный врач – единственный источник информации для первоначального варианта алгоритма. Этим должно определяться всё поведение разработчика: эксперт необходим и должен это чувствовать. Он может допускать нелогичности, противоречия, может временами не понимать, чего от него хотят, - всё это не имеет значения, не должно ставиться ему в упрёк, благо инициатива всё равно принадлежит разработчику и логичность изложения – это его забота.

Не вынуждайте эксперта терять время. Намеченные встречи могут отменяться или переноситься только по его инициативе. Никогда не упрекайте его за ту же инициативу – он врач, за ним – больные, там его долг и обязанности.

Относитесь внимательно к случаям, когда эксперт меняет свои утверждения – сам или, тем более, под вашим давлением. Когда это происходит, меняйте свою позицию и начинайте защищать то, от чего эксперт только что отказался. Этот приём не только позволяет проверить обоснованность перемен – он ещё показывает эксперту, что его суждениями дорожат даже тогда, когда он почему-то вознамерился от них отказаться.

Работая с одним экспертом, не привлекайте к той же теме другого. Только сам эксперт может показывать черновые варианты тому специалисту, которого он считает для себя авторитетом. Кроме того, он вправе привлечь к делу помощника. Роль такого помощника состоит либо в советах после встреч с разработчиком, либо в том, что на этих встречах эксперт и помощник сменяют друг друга. Однако от встреч с ними обоими вместе разработчик должен отказываться. За упомянутыми исключениями, показывать алгоритм действий врача на этапе разработки третьим лицам не следует.

Другое дело – завершённая работа. Её и можно, и полезно показывать другим врачам (с ведома и, желательно, с участием эксперта). Возникающие при этом замечания по существу медицинской темы могут адресоваться только эксперту, но в любых серьёзных спорах участие разработчика необходимо. При этом методически правильно, чтобы разработчик не становился сразу на сторону нового предложения: надо попытаться отстоять то, что уже сделано. За уже сделанным – опыт тщательного обсуждения и привязки к точно определённым этапам лечебного процесса, тогда как новшества предлагаются сходу, без такой серьёзной проработки.

Не поступайте принципами и функциями автоматизированной системы. Если алгоритмы действий врача будут использоваться, то, несомненно, только в автоматизированном варианте. Однако тот программный комплекс, который сделан мною, сегодня выглядит устаревшим. На это немедленно обратит внимание не только любой программист, но и всякий, кто работает с компьютером. Правда, несовременный вид не меняет того факта, что этот продукт доступен каждому, удобен и работает, не требуя ничего, кроме Windows, на любом компьютере, ноутбуке или планшете. Ко всему этому он бесплатен. На мой старорежимный взгляд, этого для врача и разработчика достаточно.

Но программисты смотрят на вещи иначе и непременно будут убеждать врачей в необходимости приспособить к делу смартфон, облака, веб-интерфейс, связь с медицинскими справочниками, другой дизайн и т.д., - короче говоря, всё, что каждый день обрушивает на нас технический прогресс. И надёжность надо повысить, и информацию защитить, и с медицинскими справочниками в Интернете связать.

Я не спорю. «К чему напрасно спорить с веком? Обычай – деспот меж людей». Но те контакты с программистами, которые были у меня сравнительно недавно, побуждают сделать несколько предостережений. Во-первых, надо отвергать такую реакцию на «старомодность» программы, которая выражается запелляционным заявлением, что «это всё надо переделать», и даже готовностью с ходу взяться за переделку: врачу надо всё представить по-другому, в виде схем со стрелками, программу-редактор можно сделать «совсем просто», потому что для организации диалога уже имеются готовые конструкторы, алгоритмы с помощью веб-технологий надо держать на удалении, в одном месте, доступном для всех и даже для самих пациентов. Недостатка в идеях, высказываемых с порога, нет. Есть незнание предмета, непонимание прикладной задачи.

Внешняя простота действий (задать вопрос - предложить ответы на выбор – запомнить ответ - выдать рекомендацию) вводит программиста в заблуждение. На самом деле, процесс сложен, его нельзя подогнать под уже существующие примеры и воспользоваться готовыми конструкторами или аналогами из Интернета (сегодня программисты это широко используют). Чтобы автоматизировать принятие медицинских решений, надо вникнуть во все варианты и нюансы этого процесса. Программист же, как правило, вникать не собирается. И правильно делает, это не его функция, здесь нужен компетентный постановщик задач, разработчик. А дело программиста - следовать за ним, не трогая без оснований те программистские решения, которые в уже готовом комплексе учли именно варианты и нюансы работы врача.

Что вообще может всерьёз побуждать к переделке программного обеспечения? Во-первых, условия его эксплуатации в сочетании с другими медицинскими информацион-

ными технологиями. Эти другие создаются уже на современной технической основе. Естественное желание - всё иметь в одной и той же среде, «на одной платформе».

Во-вторых, современные технические средства радикально расширяют возможности применения полезных программ и их совершенствования. Но для этого они тоже должны быть выполнены современными средствами уже потому хотя бы, что программы перестают пользоваться старыми. Так что странно было бы не предусматривать перспективу повторения программного комплекса «Частные алгоритмы действий врача» на другой технической основе. Предусматривать надо, но я решаюсь утверждать, что во всех функциях и даже в интерфейсе этот комплекс надо будет именно повторить, а не переделывать. Можно добавить новые функции, можно изменить дизайн (шрифты, цвета, наполнение экрана), можно усовершенствовать печатание протокола, но и только. Остальное надо оберегать.

Главное, что надо защищать от переделок, - это устройство базы знаний. Оно основано на понятии ситуации, воплощением которого является специально организованный dbf-файл. Я не могу себе представить другое столь же точное и экономное вместилище для отображения интеллектуальной работы лечащего врача, которое одновременно доступно пользовательской программе для автоматизации принятия решений о пациенте.

Не вижу я необходимости и в каких-либо изменениях архитектуры базы знаний, отражающей разделение медицины по специальностям с учётом рабочего места, а внутри каждой специальности – разделение тактики врача в зависимости от ведущих синдромов. Такая группировка файлов-ситуаций в виде алгоритмов и классов соответствует современной организации медицинской помощи и условиям, в которых она осуществляется. Она необходима врачу для быстрой ориентировки, без неё не может обойтись и разработчик. Здесь нет оснований что-либо менять.

Против чего нет принципиальных возражений, так это против попыток переписать с привлечением современных средств пользовательскую программу и редактор файлов-ситуаций. К таким попыткам могут побуждать не только личные пристрастия программиста, но и объективная необходимость: сопряжение алгоритмов с другими медицинскими информационными технологиями, задачи анализа накапливаемой базы данных, обеспечение защиты информации. Заметьте, что такие задачи являются внешними по отношению ко всем функциям, которые заложены в моих программах. Поэтому уже имеющиеся функции не должны быть задеты добавлением новых.

Функции моей пользовательской программы и редактора полностью отражены в их меню. Из этих функций, вероятно уже никому не понадобятся специальные средства архивирования данных и восстановления утерянных данных, а также средства авторизации

пользователя. Всё остальное относится только к деятельности врача, а в редакторе – только к описанию этой деятельности, к защите от формальных ошибок, к программной проверке сделанного. Эти функции необходимы, достаточно отлажены, и совершенствовать в них нечего. В любом новом варианте программ они должны осуществляться с теми же удобствами, полнотой и надёжностью.

ЗАКЛЮЧЕНИЕ

Что же это за занятие – разработка алгоритмов действий врача? Кто такой разработчик? Почему эксперты, опытные врачи, погружённые в свою специальность, терпят его, более того, нередко с интересом относятся к встречам с ним? Он не специалист, даже не дилетант и не всегда понятлив, но назойлив, задаёт то банальные, обывательские вопросы, от которых, тем не менее, нельзя отмахнуться, а то такие, на которые профессионал почему-то не решается ответить сразу, без тщательного обдумывания. Он дотошен, придирается к словам, возвращается к уже пройденному со своими предложениями, со своей откуда-то взятой (и на поверку – хорошей врачебной) логикой, а иной раз позволяет себе ещё и спорить. При этом практическая ценность результата этого странного общения знающего с незнающим никому заранее не известна. И всё же нечто привлекательное для врача тут есть.

Поначалу это сам факт пристального внимания ко всем деталям его интеллектуальной деятельности, к его личному поведению в профессии. Затем впечатление производит превращение высказанных мыслей в строгую логическую конструкцию. Но главное, в диалогах с разработчиком специалист обнаруживает возможность глубже осознать собственный опыт. Осознать с таким расчётом, чтобы передать его другим не в виде общих принципов и не путём показа («делай, как я»), а во всех существенных деталях, которые вдруг становятся свойствами рабочего инструмента, и этот инструмент теперь можно вручить каждому врачу. Алгоритмы излагают профессиональные знания, преломлённые, обогащённые, пронизанные взглядом практика, его логикой, врачебным опытом. Это не может не привлекать. Чувствовать себя знающим и опытным – это хорошо, но ещё большее удовлетворение доставляет такое изложение накопленного тобой опыта, которое будет служить другим врачам уже независимо от тебя.

Вот это естественное и подспудное желание каждого профессионала и реализует разработчик алгоритмов. Современной медицинской практике недостаёт детальной карты известных, открытых, исхоженных опытными врачами путей. Начальные пункты известны всем – это симптомы заболеваний. Цели, пункты назначения вечны: это выздоровление,

снятие обострения, улучшение, облегчение. А вот сказать что-то подобное о способах движения от отправных точек к цели нельзя.

Пути врача к цели, способы достижения искомого результата многообразны, время от времени они уточняются, совершенствуются, пополняются новыми открытиями. Они всегда многовариантны (а варианты большей частью неравноценны), знать их все и держать в памяти способны очень немногие профессионалы. Но даже они в силу естественных ограничений, в силу «человеческого фактора» далеко не всегда выбирают тот путь, который наиболее подходит в конкретном случае. Чем многообразнее возможные пути, тем менее надёжным помощником оказывается память. Нужна заранее составленная, отдельно от памяти существующая и время от времени обновляющаяся карта путей. Об этом и заботится разработчик алгоритмов действий врача.

Он путешествует по областям и уголкам территории под названием «Врачебная практика», проходит вместе с опытными проводниками знакомые им местные дороги и тропы, давние и проложенные недавно, и всё, что ведёт к цели наилучшим, наиболее надёжным образом, заносит на карту. Он путешественник и добросовестный картограф. Результат его работы – средство, позволяющее всем врачам пользоваться проверенным опытом хороших специалистов, обогащать его и передавать этот опыт в будущее.

Зачем было всё это описывать? Для кого? Сделаются ли алгоритмы средством повседневной работы врача? Появятся ли разработчики, без которых нельзя ни прежние алгоритмы осовременить, ни новые создать? Чтобы это получилось, нужен интерес к алгоритмам у руководителей лечебных учреждений. Пока ни интереса не видно, ни условий для его возникновения. И всё-таки однажды это было. Неожиданно, на обширной территории, с неизменным клиническим результатом, на протяжении нескольких лет. Уже из одной такой необычности стоило всё это описать и попытаться понять.