**Суперпозиция фракталов**

Карев Александр Аврамович

Изменение какого-либо измеряемого параметра объекта – процесс. Процесс схож по смыслу с понятием «команда», используемым в информатике. Эта схожесть не случайна, т.к. череду процессов в техническом объекте действительно можно представить в виде упорядоченного списка команд – алгоритма. Выполнение алгоритма приводит, в конечном счете, к выполнению главного производственного процесса (ГПП).

При рассмотрении свойств алгоритмов содержание команд не имеет значения. К примеру, информатика совсем не запрещает существования «пустых» команд, как не запрещает команде заключать в себе подпрограммы. Алгоритмам присуще фрактальное строение, основу которого (т.е., самоподобную единицу) составляет подпрограмма. Глубина вложенности подпрограмм не имеет теоретического предела. Самоподобной единицей строения информационного фрактала является минимальный набор процессов, обеспечивающих протекание ГПП минимальной ТС.

Алгоритм – это система формальных правил, приводящая к решению поставленной задачи.

Данное определение не описывает отличительных признаков алгоритма, следовательно, определением не является. Кроме того, чуть ли не все слова в этом определении сами нуждаются в определениях. Понятие алгоритма является фундаментальным, то есть таким, которое не определяется через другие, более простые понятия. Чтобы отличить алгоритм, скажем, от списка правил или обыкновенной инструкции, приведем описание его отличительных свойств:

Дискретность – алгоритм представляет процесс (технологию), как последовательное выполнение простых шагов (функций). Для выполнения каждого шага требуется конечный отрезок времени, т.е. преобразование исходных данных в результат осуществляется во времени дискретно. «Простой шаг» применительно к системам оказывается весьма непростым, т.к. связан, например, с необходимостью контроля правильности протекания процесса и надо очень четко представлять, что именно при этом происходит. Это представление связано с необходимостью иметь описание процесса и управления им. В силу изменчивости Среды это описание всегда приблизительно.

Определенность (понятность) – каждый шаг алгоритма должен быть строго сформулирован (команды не должны допускать двойственного толкования). Применительно к реальным техническим объектам команды не могут быть сформулированы так же четко, как, например, в информатике.

Связанность – на каждом следующем шаге используются результаты предыдущего;

Конечность – алгоритм должен завершаться после конечного числа шагов;

Результативность - обработка изделия должна производиться за счет выполнения конечного числа шагов. Сложный процесс обработки изделия (продукта) может и должен быть представлен в виде совокупности простых процессов, каждый из которых может быть выполнен за конечный отрезок времени. Применительно к техническим объектам правильнее говорить «…за конечный отрезок времени», а не за «конечное число шагов».

Массовость - алгоритм должен быть применим для некоторого класса задач, различающихся лишь исходными данными. Применительно к техническим объектам (системам) это означает необходимость существования отработанной технологии обработки вещества, энергии или информации. Если конструктор занимается разработкой технологии, то он занимается не своим делом, т.к. разработка технологий – удел соответствующих исследовательских коллективов. Построение алгоритма (как и технологий) требует глубоких знаний в соответствующей области, связано с тщательным анализом поставленной задачи, сложными рассуждениями. Решение по готовому алгоритму не требует каких-либо рассуждений и сводится к строгому выполнению команд. В этом случае исполнение алгоритма можно поручить не человеку, а машине. А можно ли поручить выполнение, к примеру, АРИЗ-85В (Г.С.Альтшуллер) или АВИЗ-2000 (Г.И.Иванов, А.А.Быстрицкий) машине? Конечно, нельзя, иначе это уже было бы давно сделано.

Правильность – алгоритм считается правильным, если его выполнение приводит к ожидаемому результату, но (применительно к «аппаратным» алгоритмам), соблюдение соответствия команд их описаниям не может дать абсолютной гарантии выполнения главного процесса (следствие теоремы Геделя о неполноте). Выполнение даже самого совершенного алгоритма реальным устройством далеко не всегда приводит к нужному результату. К примеру, водитель автомобиля нажимает педаль тормоза, но автомобиль не останавливается – лед на дороге! Антиблокировочные схемы тормозов имеют в своей основе более совершенный алгоритм, но и он не всегда может быть выполнен.

Эффективность – применение алгоритма должно давать положительный временной результат (экономию времени).

Не существует способа определить, что делает алгоритм, если нет его описания, т.е. дополнительной информации. Данное свойство порождено существованием описаний некоторого множества команд и поставленных им соответствие действий «исполнителя» – т.е. входного и выходного языков, т.к. создатель алгоритма должен хорошо представлять, к каким последствиям приведет выполнение той или иной команды.

Как относиться к размеру алгоритма? Интуитивно понятно, что «короткий» алгоритм выгоднее «длинного», но суровая правда заключается в том, что не существует правил построения самого компактного алгоритма. Существует набор эмпирических приемов, направленных на минимизацию алгоритмов, но для алгоритмизации технологии они совершенно не годятся.

Существуют и специфические проблемы - это инициирование (запуск) алгоритма и его остановка. Проблема остановки имеет достаточно простое решение, т.к. алгоритм САМ может содержать в себе подпрограмму проверки наступления некоторого события, при котором выполнение алгоритма должно остановиться. Кроме того, выполнение алгоритма может быть в нужный момент остановлено человеком. Инициирование алгоритма может производиться человеком или автоматическим устройством, настройку которого все-таки должен произвести САМ человек. Вот и ответ на вопрос о роли человека в «симбиозе» человека и системы – минимальное участие человека может быть сведено к запуску технического объекта. Минимально необходимый для инициирования системы объем информации составляет 1 бит. Не существует способа отличить посланный человеком бит от случайной флуктуации Среды, но эта задача решается за счет усложнения процедуры инициирования.

Разработка алгоритма, назначение моментов его запуска и остановки означают наличие в создаваемых человеком системах замысла.

Алгоритм допустимо считать теорией, содержащей алфавит языка и множество истинных утверждений языка, поэтому в отношении его справедлива теорема Геделя о неполноте, которая утверждает примерно следующее: При определенных условиях в любом языке существуют истинные, но недоказуемые утверждения. Применительно к системам теорема будет звучать так: Выполнение алгоритма не гарантирует выполнения ГПП.

Главная функция ТС непосредственно связана с конструкцией рабочего органа, поэтому закон опережающего развития рабочего органа является следствием теоремы Геделя. В то же время, из теоремы не следует явным образом классическая формулировка этого закона, что, якобы, система (технический объект) должна развиваться неравномерно. Нет сомнений, что совершенствование рабочего органа может привести к изменению всего алгоритма, но невозможно предвидеть, какими должны быть эти изменения – равномерными, неравномерными или еще какими-то. Необходимость в постоянном уточнении не бесспорного главного процесса может привести к полному обновлению алгоритма и к соответствующему изменению ТС или технического объекта.

При конструировании технического объекта необходимо дать исчерпывающее описание процесса и способа управления им. Каким бы точным ни было это описание, оно всегда будет содержать спорное положение, постулат или формулу, уточнение которых способно существенно изменить смысл процесса. Точно так же, как и в случае с ГПП, можно утверждать: Даже самое тщательное следование описанию процесса не гарантирует его выполнения. Из утверждения следует:

Объективный мотив развития систем - необходимость постоянного совершенствования выполнения всех, без исключения, процессов.

Этот мотив усиливается при существовании где-то неподалеку более совершенного технического объекта. Не следует смешивать данный мотив с необходимостью устранения ошибок развития, в основе которых лежат субъективные причины – такие, например, как подмена нужного свойства введением дополнительной управляющей функции, т.е. ручного труда.

Весьма важным для понимания специфики систем является понимание термина «среда исполнителя». СРЕДА ИСПОЛНИТЕЛЯ - совокупность условий, при которых становятся выполнимыми все известные «исполнителю» команды. Отказы «исполнителя» возникают, если команда вызывается при недопустимом для нее состоянии Среды. Как бы ни были разнообразны возможности «исполнителя», они всегда ограничены!

Выполняя алгоритм, «исполнитель» может не вникать в смысл того, что он делает и тем не менее получать нужный результат. «Исполнитель» действует формально, т.е. отвлекается от содержания и смысла задачи и только выполняет в строгой последовательности все действия. Примером формального исполнителя может служить стиральная машина-автомат, которая неукоснительно исполняет предписанные ей действия, даже если в нее забыли положить стиральный порошок и/или белье. Популярные в классической ТРИЗ многочисленные АРИЗы на самом деле алгоритмами не являются. Мало того, само их существование недоказуемо, т.е. под ними нет логической основы. Дело в том, что бесконечно большое число возможных состояний Среды может вызывать столь же бесконечное число нежелательных эффектов (НЭ). Ни состояния Среды, ни их следствия, коими являются НЭ, ни методы устранения НЭ невозможно классифицировать, следовательно, невозможно создать алгоритм устранения НЭ, как бы печально это ни было. Сказанное касается только проблемы устранения НЭ, но никак не их поиска. Алгоритм поиска НЭ может быть разработан на основе дерева процессов (одно из представлений информационного фрактала). Пример такого алгоритма – см. статью «Процессный» алгоритм.

В технике используются два способа реализации алгоритмов – «программный» и «аппаратный». «Программный» способ применяется наряду с «аппаратным», но для его реализации надо располагать некоторыми специфическими средствами – носителем программы, устройством считывания, процессором и устройством вывода. «Аппаратный» способ отличается тем, что программа «зашита» непосредственно в структуре системы, а ее реализация не требует специальных устройств считывания. Нет принципиальных запретов на комбинирование «аппаратного» и «программного» способов между собой в любых сочетаниях.

Своеобразие «аппаратных» алгоритмов заключается в том, что система содержит не одного исполнителя, а несколько. Этими исполнителями являются элементы, связи и потоки, из которых построена система. В многопроцессорных компьютерных системах происходит нечто похожее – обрабатываемая информация искусственно разделяется на потоки, каждый из которых обрабатывается своим процессором, а каждый из потоков можно считать самостоятельной командой или подпрограммой. Функционирование процессора в такой системе мало отличается от функционирования любого из элементов совершенно рядовой ТС. В конце концов, всё разнообразие команд внутри процессора преобразуется так, что он, в конечном счете, занимается очень и очень примитивным делом - двоичным суммированием битов.

Энергетические потоки в системе тоже имеют вид фракталов, самоподобной единицей строения которых является ветвление. Главное требование к энергетическому потоку – сквозное прохождение от источника в Среду (Надсистему). Иногда бывает достаточным обеспечить хотя бы временное прохождение потока.

Самоподобной единицей строения вещественного фрактала, являющегося материальным воплощением алгоритма обработки вещества, энергии, информации является минимальная ТС, число которых в конкретно взятом техническом объекте определяется правилом: «одна ТС – один процесс».

Любой технический объект представляет суперпозицию (неразрывное единство), как минимум, трех принципиально различных типов фракталов – вещественного, энергетического и информационного. Минимальное участие человека в реализации алгоритма обработки вещества, энергии, информации может быть сведено к инициированию выполнения алгоритма.

Это означает конец спорам о идеальности систем!

Идеальность вещественного фрактала сводится к достижению минимума затрат вещества и снижению габаритов системы. Эти показатели вроде бы должны стремиться к нулю, т.к. формулировка ИКР – «системы нет, а функция выполняется». На деле же ограничения накладываются требованиями потребителя, которому не нужен, скажем, калькулятор размером с булавочную головку, которым он не сможет воспользоваться. Мерой вещей является человек!

Идеальность энергетического фрактала тоже диктуется этой формулировкой – расход энергии на выполнение функции стремится к нулю. Задача минимизации информационного фрактала не имеет такого простого решения, т.к. она сводится к возможностям существующих технологий обработки вещества, энергии, информации, а здесь господствует правило: «Лучшее – враг хорошего»!

Идеальность алгоритма (информационного фрактала) – понятие логически некорректное, т.к. здесь многое зависит от состояния Среды.

Вещественный, энергетический и информационный фракталы имеют различную природу, в силу чего достигаемая в каждом из них экономия измеряется различными единицами – килограммами, джоулями, битами. Нет смысла давать комплексную оценку идеальности системы. В конечном итоге, оценка потребителя определяет – быть или не быть системе. В то же время, представление потребителя о идеальности может существенно меняться, скажем, под воздействием рекламы.

Имеет смысл говорить о оценке идеальности найденного варианта решения. Такой оценкой может служить экономическая эффективность, т.к. и килограммы, и джоули, и биты имеют цену – каждый свою.

От суперпозиции фракталов несложно перейти к формулировке определения «искусственной системы»:

Искусственная система есть материальное воплощение алгоритма (технологии) обработки вещества, энергии, информации. Объектом некоторых технологий может также являться человек, группа, население отдельной страны или группы стран.

При таком подходе полезный процесс может быть детально описан с применением общеизвестных физических эффектов, законов и математических зависимостей. А в чем же тогда роль и место ТРИЗ?

Место ТРИЗ – прогнозирование развития с учетом бесконечно большого разнообразия факторов Среды. Главная проблема заключается в определении необходимости учета этих факторов. Есть стабильные факторы, которым система изначально должна уметь противостоять – например, поле тяготения. Есть нестабильные факторы – например, температура Среды, освещенность, движение воздуха, осадки и т.п. Максимум их практически ничем не ограничен и всегда существует вероятность наступления такого неблагоприятного события, которых в обозримом прошлом никогда не происходило.

Убедиться в низкой достоверности прогнозов несложно на примере прогнозирования погоды. Чем отдаленнее прогноз, тем он требует большего количества информации. Не очень сложно предсказать погоду на завтра, но для более долгосрочного прогноза может потребоваться информация о состоянии погоды на всем земном шаре. Похожим образом обстоит дело с прогнозированием развития систем, но не анализировать же для этого всю историю развития техники! Образцом для подражания может служить живая природа, в которой реализуется многообразие видов. Аналог в технике – многовариантность исполнений. На основе выявленных возможных направлений развития систем – стремление на микроуровень, различные приемы динамизации и т.п. - разрабатывается некоторое число исполнений системы (скажем, автомобиля), из которых потребитель по известным ему одному критериям отбирает для себя нужный.

Природные системы не рассматриваются в данной статье, но идея суперпозиции фракталов и в этой области может оказаться плодотворной.