

# Нарротивные представления информационных процессов<sup>1</sup>

В.В.Александров, С.В.Кулешов

*Институт информатики и автоматизации, Российская академия наук,  
Санкт-Петербург, Россия*

Поступила в редколлегию 17.05.2004

**Аннотация**—Компьютер 50-х годов и его архитектура возникла из специфической потребности определенных функциональных вычислений, и все попытки использовать их в других областях приводили к неудачам. Технология цифровой обработки сигналов в начале своего развития была цифровым воспроизведением аналоговых технологий теории связи. В данной работе предлагается переход к пониманию необходимости обработки информации, представленной в цифровом виде, вводится понятие нарротивного представления как передачи данных с помощью описания их структуры или способа их формирования.

## 1. ВВЕДЕНИЕ

Современные инженерно-технические инфраструктуры телекоммуникационной связи предоставляют не до конца осознанные возможности информационного обмена и управляющего воздействия на общество.

В работах [1,2] была сформулирована математическая модель цифровой обработки данных – основа нарротивного представления. Концептуально это восходит к понятию алгоритмической сложности, предложенному А.Н. Колмогоровым.

Новые физические и структурные решения инфотелекоммуникаций должны поставить во главу угла специфику информационного восприятия, воздействия и создания информационных потоков человеком. В работе “Семантический анализ аудиовизуального потока данных на основе адаптивно-динамической сегментации” был отражен предложенный и разработанный с 1983 года рекурсивный (фракталоподобный) подход построения математической модели, семантического анализа аудиовизуальных информационных потоков, ориентированных на цифровые технологии обработки аудиовизуальных сигналов. Исследования психофизиологических характеристик показали специфику построения и восприятия семантических компонент, их принципиально рейтинговый характер распределения, что делает сигнал либо речью, либо музыкой, либо изображением лица, либо пейзажем, то есть узнаваемым образом, на основе предварительно сформированного знания, а не копией похожей на оригинал, в соответствии с критерием среднеквадратической ошибки, то есть аппроксимации на основе спектральных преобразований.

Приведем иллюстрацию необходимости такого подхода из [3]. Итак, рассмотрим изображение на рис. 1. Данное изображение было создано с помощью программы Macromedia Flash. Его представление в виде битовой матрицы размером 550 на 400 пикселей требует 660 кбайт памяти. С точки зрения теоремы Котельникова данное представление не является избыточным, так как в изображении имеются резкие границы между элементами, дающие вклад в высокочастотные составляющие спектра изображения. Поэтому при снижении пространственного разрешения ухудшается четкость изображения. Положим, что данное изображение необходимо передать по каналу за 1 секунду. Тогда для его передачи без потерь необходим канал с пропускной способностью не менее 660 кБ/с.

<sup>1</sup> Данная статья является продолжением работ по государственному контракту 10002-251/ОИТВС-02/096-110/030603-478

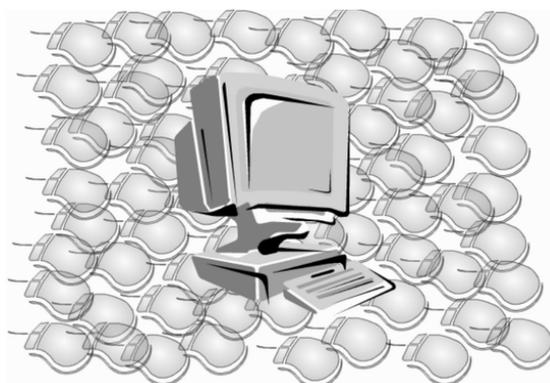


Рис. 1. Пример изображения

В то же время, поскольку изображение не содержит мелких деталей, в нем имеются достаточно сильные корреляционные связи, так что данное изображение обладает значительной вероятностной избыточностью. Эту избыточность удаляет программа-архиватор RAR, размер сжатого этой программой файла составляет всего 52 кбайт. Однако программа, формируемая системой Macromedia Flash для воспроизведения данного изображения, имеет размер всего 6 кбайт. Таким образом, на самом деле для передачи изображения необходим канал с пропускной способностью 6 кБ/с. Чем же обусловлено такое снижение требований к пропускной способности канала? Оно обусловлено семантической избыточностью, отличающейся по своей природе от вероятностной. Наличие этого вида избыточности определяется существованием в изображении закономерностей, отличающихся от вероятностной модели сигнала. Для их рассмотрения необходимо использовать иную математическую модель представления данных. При этом существующая теория связи основана на вероятностной модели сигнала.

В данной работе предлагается переход к пониманию необходимости обработки информации, представленной в цифровом виде – последовательностями единиц и нулей.

## 2. ИСТОРИЧЕСКИЙ ЭКСКУРС

Наиболее распространенными являются следующие типы информационного восприятия человека:

- звук;
- изображение;
- текст.

Эти три типа информационного восприятия проистекают из физиологических свойств разделения потоков по органам чувств (зрение, слух), а позже от постепенной замены некоторых концептов, понятий знаками (появлением языка и письменности).

Это подтверждается также принципом, который обеспечивает технический прогресс, по Дж. Тойнби [4]: *... В истории письменности наблюдается не только соответствие между развитием техники письма и упрощением формы, но и эти две тенденции фактически тождественны друг другу, поскольку вся техническая проблема, которую должно решить письмо как фиксатор, посредник человеческой речи, – это отчетливая репрезентация широчайшей сферы человеческого языка с максимальной экономией визуальных символов, т.е. этерификация – есть закон прогрессирующего упрощения. ...*

Человек заменяет знаками понятия, когда он уверен, что его адресат сумеет идентифицировать адекватное понятие по этим знакам. Если такая замена невозможна (по причине недостаточности зна-

ния, отсутствия понятий для данной области знания) или энергетической неэффективности передачи знаковой информации, передается звук или изображение.

Эти типы информационного содержания в последствии были формально перенесены на телекоммуникационные системы без дополнительного переосмысления и без учета специфики цифровых технологий и организации вычислительного процесса – алгоритма.

В компьютере все виды информационного содержания представляются в виде конечных последовательностей битов – или их объединений – знаков, идентификаторов. То есть любая информация в компьютере представляется в предельно упрощенной абстрактной форме, совокупностью “0” и “1”. А знаковое представление эффективно только при использовании вербальной составляющей, формирующей информационное сообщение как текст, язык и смысл, то есть семиологической модели информационного сообщения [7].

Концептуальное различие в компьютерной и семиологической модели информационного содержания и определяет сложность поиска алгоритмов соответствия их информационным представлениям. Подчеркнем, что основная сложность состоит не в алгоритмах и форматах компьютерного представления, а в их адаптации для восприятия информационных потоков человеком. Этим объясняются трудности алгоритмической реализации, казалось бы, простых операций над некоторыми типами представлений (например, поворота растрового изображения).

Другие типы информационного содержания (например, сигнал (физический уровень) или кардиограмма (логический уровень) даже будучи переведенная в воспринимаемый человеком вид) не рассматриваются, так как они не воспринимаются органами чувств человека и, соответственно не имеют для него смысла.

При информационном взаимодействии выделим следующие уровни:

- физический уровень;
- логический уровень;
- семиологический уровень.

Протоколы физического уровня обеспечивают непосредственный доступ к среде передачи данных. Данные передаются протоколами данного уровня в виде битов (для последовательных протоколов) или групп бит (для параллельных протоколов). На данном уровне определяются набор сигналов, которыми обмениваются системы, параметры этих сигналов – временные и электрические и последовательность формирования этих сигналов при выполнении процедуры передачи данных. Кроме того, на данном уровне формулируются требования к электрическим, физическим и механическим характеристикам среды.

Протоколы логического уровня выполняют задачи, начиная с обеспечения передачи данных по физическому носителю, обслуживания сессии и заканчивая алгоритмами преобразования формата представления данных.

Особую роль при этом выполняют протоколы представления данных – форматы. Эти форматы определяют тип передаваемой информации. Примерами таких форматов являются кодировки текста (KOI, WIN-1251), изображения (Bitmap, JPEG), звука (PCM, MPEG).

Семиологический уровень должен обеспечивать однозначную идентификацию концепта, который передавался посредством системы коммуникации.

При этом влияние семиологического уровня на логический уровень может оказаться весьма полезным, например, при передаче потенциально избыточных данных, допускающих сжатие. Практическое использование архиваторов показало наличие зависимости эффективности алгоритма сжатия от типа данных.

При этом существует ряд протоколов, объединяющих сразу несколько уровней. Так код Морзе, определяя как правила сопоставления последовательности знаков двоичного алфавита (точек и ти-

ре) знакам другого алфавита, так и некоторые параметры передачи этих символов по линии связи, является протоколом и объединяет семиологический и физический уровни.

Физический и логический уровни достаточно хорошо изучены и описываются разными моделями (например, известной моделью OSI, стандарт ISO 7498). Эти эталонные модели, не являясь реализацией, определяют функции каждого уровня. Однако модели этих стандартов требуют пересмотра и добавления дополнительного уровня.

Физический уровень – совокупность технических средств для хранения и передачи данных. Преобразования на физическом уровне и современные протоколы данных учитывают лишь аппаратную реализацию и связаны лишь с оценкой битовой составляющей по Шеннону. Этот уровень принципиально не учитывает семантико-смысловую составляющую сигнала. Результат работы на физическом уровне – это создание проблемно-ориентированных сред. [5]

Исторические этапы понимания необходимости введения семиологического подхода можно представить следующим образом.

Исторически первым инструментом информационной коммуникации людей служили проявления искусства: естественный язык, живопись, музыка. Позднее математики ввели число и алгебраические структуры. Аристотель в своих работах ввел логику. Но с появлением логики появилась и проблема – различные высказывания приводили к одинаковому принятию решения, то есть смысл (принятие решения идентификации по этому описанию) этих высказываний обеспечивал однозначную идентификацию объектов.

Компьютеры, используя логику Аристотеля в качестве формализма, унаследовали проблему идентификации, которая для человеческого мозга не так актуальна. Также, в компьютере понятие замкнутости, приносимое с алгебраическими структурами, не вносит ничего, кроме противоречий. Так, например, деление целого (integer) числа на целое дает число с плавающей точкой (float) – то есть компьютерные операции не являются замкнутыми. Некоторые математические аксиомы оказываются не выполняющимися в компьютере, например результаты вычисления  $z * (x/y)$  и  $(z/y) * x$  для компьютера, в общем случае, не равны. В рамках компьютера нет больше различий между счетными и несчетными множествами, то есть вещественное представление чисел дискретно, несправедлива теория предела, интегральное и дифференциальное исчисление.

Удачно сложилось, что математические модели дали возможность описания физической реальности. Получилось очевидное преимущество в прямом применении формализованных математических моделей к изучению и описанию физической реальности. Позднее, с Эйнштейна, началось расхождение физики и математики, что привело к необходимости физикам подстраиваться под математические законы, а математикам разработать новую математику, объясняющую физические процессы.

Среди математиков возникает идея информационного обмена. Появляется “вторая математика” – обмен объектами, которая восходит к теории множеств Кантора как взаимнооднозначного соответствия и принципу идентификации неразличимости Лейбница.

Однако конструктивно для работы компьютерной математики только А.Н.Колмогоров связал эти процессы (обмен объектами) в понятие алгоритмического содержания.

В работе [6] в понятии алгоритма входные данные и результат подменяется номером элемента упорядоченного множества входных данных и результатов соответственно. Этим понятие алгоритма сводится к операции получения идентификатора результата по идентификатору входных данных, то есть вычислению.

В дальнейшем, с развитием цифровых инфотелекоммуникационных технологий, приходится иметь дело только с информационными объектами. При этом приходится различать точное “физическое” представление объекта как непосредственное точное представление сигнала цифровыми отсчетами. Так лучшим описанием изображения для задач идентификации и принятия решения является описание изображения морфологическими способами, начиная с описания общих объектов с последую-

шим уточнением деталей, то есть переходу к описанию мелких деталей. При этом принятие решения идентификации будет происходить ранее, чем вхождение в автоколебательный (возможно, бифуркационный) процесс, вызванный излишней точностью представлений, который сделает невозможной задачу идентификации.

Но информационные объекты по ориентации на человека обязаны соответствовать психофизиологическому состоянию [7], то есть соответствовать первоначальным средствам коммуникационного взаимодействия между людьми, и, прежде всего, естественному языку. Это соответствие автоматически дает морфологическое описание объектов, что позволяет повышать эффективность задач идентификации.

Существующие методы, претендующие на обработку информации, имеют дело с обработкой лишь энергетической составляющей сигнала, несущей эту информацию. Поэтому и процесс минимизации информационного описания (в смысле Шеннона) не достигает минимально возможного уровня, который достаточен при алгоритмическом или семиологическом описании информационного содержания.

Неэффективное представление данных, как правило, выражается в комбинаторном увеличении сложности операции идентификации.

Но, даже работая в энергетическом представлении, необходимо учитывать особенности реальных данных, организовывать самоподобные структуры и использовать для их хранения и передачи методы на рекурсивной основе [2]. Наличие самоподобия в реальных данных наглядно демонстрируется бурно развивающейся областью создания архиваторов, работа которых основана на удалении избыточности, за счет создания “словарей” и других способах.

В простейшем случае без изменения форматов представлений для реальных данных требуется введение эвристического анализа для определения типа информационного содержания и применения соответствующего алгоритма протокола логического уровня (например, компрессии или выбора физического канала, оптимально соответствующего передаваемым данным) на основе семантической составляющей.

Так для текстовых данных, возможно наилучшим может быть код, подобный коду Морзе, объединяющий в себе семиологический и физический уровни, а также отражающий статистические свойства передаваемых данных. Аналогично, в аудиовизуальных форматах, подобных MPEG, важно представить данные не как коэффициенты преобразования в некотором математическом формализме, а как описание действий последовательного перехода из одного состояния в другое.

### 3. НАПРОТИВНОЕ ПРЕДСТАВЛЕНИЯ. ДАННЫЕ – ПРОГРАММА – СИНТАГМА

Компьютер 50-х годов и его архитектура возникла из специфической потребности определенных функциональных вычислений, и все попытки использовать их в других областях приводили к неудачам. Технология цифровой обработки сигналов в начале своего развития была цифровым воспроизведением аналоговых технологий теории связи.

Переход к новому этапу развития цифровых технологий обработки сигналов в первую очередь проявляется в отказе от доминирующего в аналоговой технике математического аппарата, построенного на теориях дифференциального и интегрального исчисления, функций комплексного переменного, рядов и интегралов Фурье.

Вместо него следует использовать математический базис, основу которого составляет алгоритмический подход к теории информации, синоним которого – вычислимые функции [6], машина Тьюринга, алгоритмы Маркова [1,8].

Традиционный перенос математических построений на компьютер вызывает трудности в задачах, не являющихся математическими, а особенно в задачах, связанных с задачами идентификации, классификации, искусственного интеллекта. Раньше компьютер понимался лишь как интерпретатор записи чисел. Его архитектура и парадигма программирования строились именно для решения этой

задачи, но в результате из компьютера получилось нечто другое – и сейчас пришло время понять, что именно из этого получилось.

Нарротивное представление (от лат. *narrō* – повествовать) – способ представления и передачи данных с помощью описания их структуры. То есть передается не сам объект, а описание способа его формирования.

Принципиально сам метод не является новым. Понимание этого уже проявляется в реализации, например, визуализационных компонентов, исполняющихся на стороне клиента (технология Macromedia Flash и Java Applets). В простейшем случае (хотя возможности Flash и Java этим не ограничиваются) вместо элемента мультимедийного содержания (картинки, звукового фрагмента) передается исполнимый код, выполняющийся на компьютере клиента. Их исполнение, как правило, происходит в среде виртуальной машины.

Требуется лишь обобщить понятие нарротивного представления на разные виды существующих форматов данных и определить направление развития новых форм представления данных.

Для дальнейших рассуждений необходимо отличать понятие “число” от понятия компьютерно-вычислимого числа как объекта. В отличие от “числа” в математическом понимании, “компьютерное число” суть идентификатор – уникальное состояние, которое может использоваться для идентификации объектов реального мира и их свойств, который по счастливой (а возможно и нелепой) случайности напоминает по свойствам (возможности обозначения порядка и количества) число в математическом понимании. При этом математика определяет необходимые для построения модели зависимости. Здесь начинается расхождение математического и компьютерного чисел – для компьютерных чисел, как идентификаторов объектов (в том числе и как идентификаторов математических чисел) не обязательно выполнение условий, накладываемых алгебраическими структурами, и, в первую очередь, условия замкнутости.

К понятию программы также существует несколько подходов. В работе [6] в понятии алгоритма входные данные и результат подменяет номером элемента упорядоченного множества входных данных и результатов соответственно. Этим понятие алгоритма сводится к математической операции по вычислению идентификатора результата по идентификатору входных данных, то есть к вычислимой функции, которая в качестве идентификатора используется число.

При этом математические свойства чисел, в общем случае не требуются, но различные технологии программирования, как правило, используют свойства чисел.

Другой подход основан на принципах машины Тьюринга и машины Поста. В данной парадигме программой является описание дискретных действий над объектами, являющимися данными.

Программа обычно рассматривается как некоторая запись существующих взаимосвязей в выбранной предметной области. Чаще всего это запись некоторого алгоритма в выбранной нотации (процедурная или декларативная), но возможны случаи, когда программа не является записью никакого алгоритма, поскольку в записи программы есть неразрешимые противоречия. Например, бесконечный цикл без возможности выхода есть запись требования бесконечного вычисления приближения к результату без необходимости когда-либо остановиться. Исходя из этого понимания, программа – это способ интерпретации записи числа.

Следуя подходу Колмогорова [6] об алгоритмическом описании, введем понятие терминальной программы.

Назовем терминальной программу, которая формирует выходной результат – последовательность (или агрегат) идентификаторов. Таким образом, терминальная программа является вырожденной программой без входных данных. Терминальная программа является самодостаточной, то есть ей не требуется ничего, кроме исполнителя, работающем в некотором формализме (процессора).

Необходимо различать два типа программы, которые преобразуют номер исходных данных в номер результата и исходные данные непосредственно в результат.

В парадигме цифровых технологий стирается различие между понятием программы и данных. Программа, являясь последовательностью команд, хранимых как в виде байт-кода (кодов операций процессора) так и в виде исходного или объектного кода на некотором языке является обычными данными, которые могут обрабатываться другими программами.

Важно понимать любой формат, не как математическое представление, а как необходимую добавку к алгоритму, чтобы он стал терминальной программой, то есть восстановил исходный сигнал. В этом случае стирается граница между программой и данными. То есть данные в некотором формате являются кодом для выполнения универсальной программы порождения данного вида объектов (или универсальной функции). Это означает, что универсальная функция дополняет формализм исполнителя, то есть становится сама частью процессора для выполнения некоторой терминальной программы.

Некоторые типы данных допускают непосредственное представление. Как правило, такие данные являются данными в какой-либо знаковой форме, например текстовые данные.

Непосредственным представлением называется формат данных, не требующий дополнительного преобразования для выдачи данных потребителю. Как правило, все такие форматы являются полным описанием некоторых физических величин, описывающих объект.

Если рассматривать некоторый сигнал как конструктивный объект (то есть объект, который может быть построен вычислимой функцией), порождаемый терминальной программой, то величина алгоритмической избыточности сигнала определяется разностью между объемом цифрового представления совокупности его отсчетов и длиной программы его порождения. Доказательство существования конструктивного объекта не требуется.

В тех случаях, когда сигнал не содержит закономерностей, минимальной программой его порождения является простое перечисление его отсчетов. В этом случае длина программы совпадает с объемом цифрового представления отсчетов сигнала, а алгоритмическая избыточность равна нулю.

Можно рассмотреть компьютерную программу воспроизведения звуковых файлов как универсальную функцию, способную сформировать произвольный выходной звуковой сигнал. Любой конкретный звуковой файл конкретизирует эту функцию до терминальной программы, порождающую конкретный звуковой сигнал.

Передачу данных можно определить как сведение задачи сложности NP, то есть получения решения перебором без знания алгоритма решения, к задаче сложности P. Получение дополнительных данных позволяет доопределить алгоритм до возможности полиномиального решения задачи. Наглядный пример такого доопределения продемонстрирован в [9]. В этом методе производится гаммирование (шифрование смешиванием полезного сигнала с шумоподобным сигналом). Попытка восстановления исходного сигнала при знании алгоритма формирования (программы) гаммирующего сигнала, невозможно более эффективным методом, чем перебор по всему множеству параметров, задающих конкретный сигнал. Знание этих параметров сводят задачу восстановления полезного сигнала к задаче сложности P, то есть простое алгоритмическое восстановление.

Предположим, что:

- нам известен универсальный алгоритм порождения сигналов, такой, что любой сигнал может быть порожден с помощью программы, длина которой меньше, чем объем цифрового представления совокупности его отсчетов;
- имеется процессор неограниченной производительности.

Тогда можно построить процедуру подбора программы под заданное изображение. Теоретически, это может быть прямой перебор, так как имеется процессор неограниченной производительности.

В реальных условиях производительность процессоров ограничена, и это потребует поиска более эффективных процедур подбора программ, порождающих заданные сигналы. Однако в теоретическом плане можно утверждать, что существует возможность переноса части требований по ресурсам

каналов передачи данных на требования по ресурсам процессоров, осуществляющих обработку данных на входе и выходе каналов.

Следовательно, рост производительности процессоров, снижение их стоимости и габаритов дают возможность уйти от традиционных ограничений по пропускной способности для передачи, хранения и обработки сигнала, и создать новое поколение недорогих устройств для получения, хранения и передачи высококачественного видео и звука.

К сожалению, число как идентификатор программы, соответствует бесконечно большому количеству программ для различных интерпретаторов. Для любого множества можно определить алгоритм, сопоставляющий число элементу этого множества и наоборот, то есть функцию нумерации элементов. То есть любой элемент любого множества при выбранном для него алгоритме будет записью некоторого числа. То есть программе тоже можно будет сопоставить некоторое число, и при такой трактовке можно определить программу как результат интерпретации записи некоторого числа.

Такая подмена данные-программа-данные изменяет концепцию передачи данных: передаваться могут не только последовательности данных (то есть последовательности компьютерных чисел), но и терминальные программы, порождающие эти данные (то есть описания этих данных) также в виде компьютерных чисел.

Основная отличительная особенность нарротивного представления от физического представления заключается в использовании, в какой-либо мере, семантической – семиологической компоненты данных. Архивирование с компрессией, использующее статистические свойства данных также может быть отнесено к нарротивному представлению. Собственно данными по этому являются только непосредственные представления.

Если формат данных не являются непосредственным представлением, то не имеет значения, чем считать данные в таком формате – собственно данными или терминальной программой. Исполнителем в этом случае может являться не интерпретатор машинных кодов, а транслятор, преобразующий значения данных формата в выходной сигнал.

Переход к пониманию любых форматов представления как нарротивных кодов, позволит избежать промежуточных представлений в формализме некоторой математической модели.

Приведем наглядный пример. Пусть имеется реализация универсальной функции, способной формировать любое изображение. Аргументом этой функции является уникальный идентификатор соответствующего изображения. Получается, что функция “умеет” строить любые изображения, а пользователь лишь указывает номер требуемого изображения. Если некий художник создаст новое изображение, то ему достаточно будет сообщить лишь имя этого изображения – его уникальный идентификатор, чтобы другие пользователи смогли его увидеть, введя в качестве аргумента функции этот идентификатор.

В настоящее время эта идея реализована через визуализацию изображения, записанного в некотором формате. То есть содержимое некоторого графического файла и является уникальным идентификатором этого изображения. Но при этом этот идентификатор содержит избыточный объем данных, который возникает из модели представления этого формата.

Понимание этого дает возможность создавать форматы, которые не базируются на промежуточных представлениях, а являются собственно идентификаторами.

Понятно, что любой физический канал обладает, строго говоря, ограниченной пропускной способностью. Но так как в парадигме цифровых технологий становится возможной передача как непосредственно самого сигнала, так и его более компактного описания, достаточного для ряда задач, то традиционное понимание пропускной способности, применительно к передаче информации в цифровом виде, становится некорректным.

Полная пропускная способность канала требуется только в том случае, когда передаваемый сигнал представляет собой реализацию нормального белого шума. В реальных сигналах всегда имеются

отклонения от этой модели, то есть закономерности, которые позволяют осуществить цифровое сжатие сигнала перед подачей в канал и восстановление сигнала на выходе канала. Поэтому процессор компрессии сигнала способен уменьшить требования к пропускной способности канала на величину, определяемую найденными в сигнале закономерностями.

Следовательно, в зависимости от эффективности метода компрессии требования по пропускной способности канала изменяются. При этом теорема Котельникова задает предельное значение пропускной способности при стремлении к нулю эффективности компрессии.

Традиционное программирование предполагает необходимость задания на множестве чисел отношения, отражающее свойства реального мира, а программа, согласно определению – запись соотношений (свойств) реального мира. Но в компьютере программа (а также и терминальная программа) не отражает ничьих свойств, она только формирует выходные данные.

Из этого положения следует вывод, что программа не должна строиться из соображений имитации реальных явлений (хотя это и привычная и кажущаяся единственно возможной технология программирования), а формироваться из условий входа – выхода (скорее всего автоматически). Пример постепенного понимания этого факта виден в переходе от программирования в кодах к программированию на языках высокого уровня, то есть сгенерированные коды делают уже на каждом шаге не то, что хочет человек, если бы он формировал (программировал) их вручную, но, тем не менее, выполняют общую задачу, реализованную на языке высокого уровня.

Получается, что существуют два принципиально различных способа создания программ: традиционное (аналитическое) создание как описание явлений реального мира или математической модели физического мира, и автоматическая генерация, основанная на реализации потребительского интереса, минуя этап математической формализации и поиска математически оптимального решения. При этом разработка ведется непосредственно в терминах предметной области без привязки к конкретному математическому аппарату.

Существование подобных эвристических подходов не снимает, однако, проблемы общего теоретического описания методов цифровой обработки сигналов, поскольку даже если разработка алгоритма может быть выполнена эвристически, без осмысления физических и математических процессов невозможно оценить предельную эффективность, области применимости, вести совместную разработку и, наконец, накапливать и передавать знания об этих технологиях.

Приведем еще один наглядный пример, иллюстрирующий процесс перехода “данные-программа-данные”. Если попросить некоторого художника нарисовать “домик под деревом”, то он нарисует картину, в которой некоторый дом будет расположен рядом с некоторым деревом. В этом случае художник будет являться интерпретатором, наши слова “домик под деревом” – будут являться терминальной программой, то есть описанием, достаточным (в рамках нашей задачи) для реконструкции реальной картины (которая выступает в качестве исходных данных). Примечательным фактом здесь является то, что последовательность слов естественного языка “домик под деревом” может быть как терминальной программой, так и данными (например, если эта фраза встречается в художественной книге). Картина тоже может быть как данными, так и программой (например, если это сборочный чертеж). Некоторые данные будут являться для человека программой, инициализирующей у него процесс формирования образа и т.д.

Предлагаемый подход, по своей сути, предполагает наличие универсальных инструментов для создания продукта. При этом нет необходимости транспортировать сам продукт, а достаточно передать лишь его описание (схему, сборочный чертеж) для воспроизводства его получателем. Инфотелекоммуникационные технологии при этом являются основой поддержки функционирования такой системы.

Разделение типов программ и введение понятия терминальной программы позволяет также получить иерархическую классификацию программ: вершиной дерева является универсальная программа, а терминальными узлами дерева являются терминальные программы. Подобная иерархия позволяет

построить систему, которая позволяет предсказывать алгоритмы и может дать эффективный способ их автоматической генерации.

#### СПИСОК ЛИТЕРАТУРЫ

1. Александров В.В., Арсентьева А.В. *Информация и развивающиеся структуры*. Л.: ЛНИВЦ АН СССР, 1984.
2. Александров В.В., Горский Н.Д. *Представление изображений. Рекурсивный подход*. Л.: Наука, 1985.
3. Цветков О.В. *Цифровые технологии обработки аудиовидеосигналов: компрессия и семантический анализ*. Труды СПИИРАН. Вып. 2. СПб: СПИИРАН, 2004.
4. Тойнби Дж. *Постижение истории*. М.: Прогресс, 1996.
5. Александров В.В. *Развивающиеся структуры и проблемно-ориентированные среды. Теоретические основы и прикладные задачи интеллектуальных информационных технологий*. СПб.: СПИИРАН, 1998.
6. Колмогоров А.Н. Три подхода к определению понятия “количество информации”. *Проблемы передачи информации*, 1965, 1, стр. 3–11.
7. Эко У. *Отсутствующая структура. Введение в семиологию*. СПб.: Симпозиум, 2004.
8. Simon J.C. *Patterns and Operators. The Foundations of Data Representation*. R. R. Donnelley and Sons Company, 1986.
9. Кулешов С.В. *Фрактальное шифрование*. Труды СПИИРАН. Вып. 2. СПб: СПИИРАН, 2004.

*Статью представил к публикации член редколлегии В.И.Венец*