

КОМПЬЮТЕРНЫЕ НАУКИ

УДК 004

© А. П. Бельтюков, С. Г. Маслов, О. А. Морозов

КОНСТРУКТИВНЫЕ МЕХАНИЗМЫ В ИТ–СФЕРЕ¹

Рассмотрены проблемы и условия, связанные с формированием конструктивных механизмов построения ИТ-сферы.

Ключевые слова: ИТ-сфера, системные описания, языковая форма и образы, конструктивная деятельность, информационный артефакт, программирование, физическая реальность, ИТ-система, либернетика, стратегии компьютинга.

В современном обществе ИТ-сфера становится системообразующим фактором высокого качества жизни, фактором оперативного и гибкого реагирования на кризисные явления, а также стратегическим направлением формирования и развития жизнедеятельности как отдельно взятой личности, так и общества в целом. Кроме того, именно ИТ-сфера должна стать катализатором и концентратором знаний высококонтрастного гармоничного устойчивого развития человека, общества и природы. Эти процессы являются прерогативой творческого подхода, когда мысль должна опережать выполняемые действия и гарантировать устойчивое развитие.

В системах ИТ-сферы наиболее ярко проявляется феномен сложности, поскольку они воплощают в себе попытки построения систем на границах человеческого знания и интеграции наших знаний и понимания в наиболее полной и массовой форме как об окружающей действительности, так и о самом человеке, и часто в критических ситуациях с неустойчивым поведением. Этот феномен сложности — основная «головная боль» не одного поколения математиков, программистов и информатиков, но именно его преодоление дает ключ к устойчивому и прорывному развитию.

Построение программ или информационных систем — это реализация наших творческих (креативных и конструктивных) способностей, реализующих смысл понимания жизни, т. е. формирование потребностей, выявление проблем и целей. Позиция творца (программиста, информатика, математика, конструктора, ...) при преодолении феномена сложности все время совершает маятниковые движения между сущностью и явлением, объектом и процессом, меняя точку зрения, акценты, критерии и цели. Более детально огромный творческий путь в ИТ-сфере можно отразить в виде потока идей и попыток их реализации:

- *универсальный язык* программирования (PL/1, Algol-68, Ada);
- *универсальная технология* программирования (нисходящее программирование, сборочное программирование, R – технология, Σ – программирование, UML – технология, ...);
- *многофункциональная среда* программирования (NUT, Delphi, Maple, ...);
- *разнообразие стилей* программирования (императивный, логический, функциональный, сентенциальный, объектно-ориентированный, автоматный, событийный, на ограничениях, тензорный, генетический, ассоциативный, приоритетный, ...);
- *кроссплатформенность* (CORBA, J2EE, NET, ...).

Это фактически непрерывное экспериментирование с различными формами, логикой и семантикой, которое неизбежно приводит к главному вопросу: что есть основание (или базис)

¹Работа выполнена при финансовой поддержке РФФИ (грант 08-07-00460-а).

и каковы механизмы построения необходимых нам объектов и процессов (эффективной реконструкции и/или создания принципиально нового)? Если рассматривать виртуальную реальность (или воображаемый мир) и физическую реальность (или реальный мир), то какова их взаимосвязь? И если под конструктивной реальностью (или искусственным миром) понимать часть виртуальной реальности, которая может быть материализована в физической реальности, то каковы законы ее формирования и эволюции (рис. 1)?



Рис. 1. Виды реальностей

Кроме того, в процессе построения систем в ИТ-сфере можно также говорить и о других трех физических реальностях (рис. 2):

- *реальность 0-рода* — сам субъект, «биокомпьютер», тот кто принимает решения, формулирует цели, задачи и критерии оценки;
- *реальность 1-рода*, та которую наблюдают и моделируют, которую преобразуют и используют;
- *реальность 2-рода* (традиционный, молекулярный, квантовый компьютер), та которую используют для моделирования и познания, в безопасных и масштабируемых условиях (в каком-то смысле эффективных).

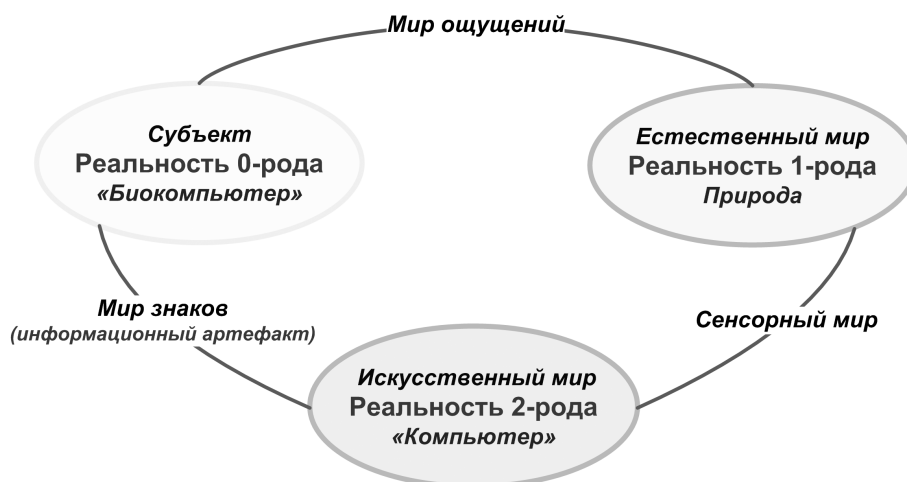


Рис. 2. Структура реальности и информационные потоки

Простого взаимодействия физических реальностей 0-го и 1-рода современному человеку уже недостаточно, поэтому возникает физическая реальность 2-рода. Симбиоз двух вычислителей (0-я и 2-я физические реальности) — это управление информационными, энергетическими и вещественными потоками, осуществляемое посредством некоторой языковой и образной формой их представления (абстрактные и конкретные знаковые формы и операции с ними).

В рамках последних трех физических реальностей, если встать на системную точку зрения [1], то основной творческий процесс — ПРОГРАММИРОВАНИЕ — можно определить как

управление информационной средой и потоками информации между физическими реальностями 0-го, 1-го, 2-го рода, в процессе которого создаются и используются информационные артефакты (образы) абстрактных и реальных систем (объекты или процессы), удовлетворяющие потребности субъекта.

Информационные артефакты строятся в рамках мира знаков на

естественных и/или искусственных языках.

Естественный язык — это форма, которая акцентирует наше внимание на понимании, ассоциативности, аналогичности и целостности наших представлений. *Искусственные языки* — это формы, которые дают более глубокие и точные представления информации и знаний, адекватные решаемой задаче, способностям субъекта, и акцентируют наше внимание на вычислимости, надежности и эффективности. Эти формы связаны процессами формализации и деформализации:

естественный язык

↔ *профессиональная проза*

↔ *терминологическая система*

↔ *язык прикладной научной теории*

• *системное описание* (рис. 3).

Важно, чтобы вся цепочка языков имела согласованную в целом систему семантик, допускающую образное представление и компьютеринг (в частном случае, вычисления, вывод, навигацию, ...) в некоторой среде 2-рода. С этой точки зрения целесообразно уточнить взаимосвязь системного анализа [1, 3] и языковой деятельности.



Рис. 3. Слои системного описания

Морфологическое описание, определяющее состав систем, соответствует синтаксису языка. Однако синтаксис, претендуя на правильно построенные конструкции, часто приводит к «невыполнимым» конструкциям, несовершенство которых проявляется в поведении.

Функциональное описание, определяющее динамическое поведение и преобразование систем, соответствует различным семантикам, например [2, 4] денотационной (исчисление конверсии объектов, динамика их агрегирования и детализации), аксиоматической (аксиомы с содержательной точки зрения делятся на постоянные (законы) и переменные (условия), правила вывода), функциональной и другим.

Атрибутивное описание, определяющее качественное и количественное проявление систем, соответствует логической семантике, описывающей свойства устройства и поведение систем, в частности их надежность, защищенность, сложностные и эргономические аспекты, которые могут быть использованы в процессе построения, выбора и эксплуатации систем.

Либернетическое описание не имеет явного соответствия в языковой деятельности. Оно концентрирует внимание на инвариантных и вариативных, проекционных и интеграционных особенностях конструкций и поведения системы. Важной особенностью такого описания является переход от базовых к обобщенным представлениям (координатам) с помощью программ управления, необходимых и достаточных для достижения целей.

Генетическое описание, обнаруживая и преодолевая препятствия и противоречия в построении систем, является основой выявления и концентрации опыта в фиксации устойчивых структур и поведения систем.

Генетическое и либернетическое описания в объединенном виде представляют собой *управление* жизненным циклом систем.

Сегодня можно констатировать, что конкретизация концептуального слоя в математический и алгоритмический часто производится в неявной форме, поскольку программисты, используя тот или иной традиционный стиль программирования, прибегают к определенной в нем математической, вычислительной или логической модели. Это сильно ограничивает возможности в построении ИТ-систем, т. к. приводит к неестественной и иллюзорной адекватности, к технологическим разрывам в процессе решения задач и использованию программистом одной стратегии или приема во всех случаях жизни. Фактически, когда выбор более адекватной математической формы нуждается в серьезной проработке, от него отказываются, т. е., грубо говоря, «уравниваются» геометрические (тензорные), логические, вероятностно-статистические и нечеткие постановки задач. Этот крайне важный аспект построения информационного артефакта требует отдельного обсуждения.

В данной статье пока сконцентрируемся на программистских аспектах. Необходимо отметить некорректное совмещение понятий «парадигма» или «стиль» программирования, которое ведет к потере ясности понятия «конструктивность», нет четкости распределения функций в рамках выделенных трех реальностей. Если исходить из того, что информационная система является симбиозом биокомпьютера и компьютера в обработке информации, то необходимо сформировать в этом отношении точки зрения на атрибуты программирования, представленные в виде таблицы (табл. 1), с пометкой «*» взаимодействующих аспектов.

С точки зрения выделенных реальностей целесообразно для программирования выделить в качестве атрибутов стили, формы, виды и техники, которые концентрируются на едином стержне — мире знаков.

Естественно, что основные проблемы и задачи возникают в головах людей (личностей или в среде субъектов) и первое, что требуется — это организовать в этих головах ясность представления и «правильный ход мыслей» об окружающем их мире и о самих себе. Организовать эффективность и развиваемость индивидуального и коллективного мышления и деятельности в процессе наблюдения, познания, конструирования и понимания действительности. Осуществляя все это с помощью идеальных, абстрактных и конкретных образов, представленных в мире знаков, а также воспринимаемых через мир ощущений, жестов и мимики. Ядром здесь является *стиль* мышления, учитывающий репрезентативную систему, интеллект, социотип, психотип, темперамент и другие факторы. Стиль этот строится на основе системной психофизиологии

Таблица 1. Аспекты программирования

| Атрибут программирования | Субъект (личность, коллектив) | Естественная реальность | Искусственная реальность | Мир ощущений | Сенсорный мир | Мир знаков |
|--------------------------|-------------------------------|-------------------------|--------------------------|--------------|---------------|------------|
| Стиль | * | * | | * | | * |
| Форма | * | | * | | * | * |
| Вид | | * | * | | * | * |
| Техника | * | | * | * | | * |

(Ю. И. Александров, В. Б. Швырков) и представлениях об уровнях когнитивной организации и регуляции поведения (Б. М. Величковский). В этом случае можно говорить как о метафорическом, моторном, мягком и жестком, абстрактном и конкретном, и других стилях, так и о концепции единства сознания и эмоций, о мире знаков, фиксирующем промежуточные и конечные результаты мышления и деятельности.

Форма является частью результата материализации замысла субъекта, которая позволяет представить информационный артефакт не только в мире знаков, но и в сенсорном мире. Можно говорить о полисенсорной, бисенсорной (мультимедийной), лингвистической, геометрической, логической и других формах, позволяющих наглядно и осязаемо представить абстрактные и конкретные проекции, а также целостное видение информационного артефакта и характеристик управления им.

Вид связан с характером преобразования и построения форм, взаимодействием естественной и искусственной реальности, с оптимизацией баланса между объектами и процессами в представлении информационных артефактов. Сейчас известны следующие виды: трансформационный, конкретизирующий, сборочный, синтезирующий, доказательный, верифицирующий, фрактальный, генетический и/или эволюционный и ряд других.

Техника связана с организационными аспектами деятельности субъекта в процессе создания и использования информационного артефакта. В качестве известных техник можно упомянуть техники проектирования, экстремальное или бригадное программирование, технику формализованных заданий и другие.

В процессе программирования и исполнения программ сущности и явления отображаются в виде объектов и процессов или понятий и их преобразований. Если следовать некоторой языковой форме, то термин, обозначающий понятие, может иметь постоянное значение (инвариант или константа) или переменное (вариации или альтернативы). Кроме того, имеются точки зрения, отражающие применимость термина в некотором контексте. Интерпретация термина может быть выражена цепочкой понятий, фиксирующей некоторый контекст преобразования или управления:

термин (объект, процесс, свойство, ...)

↔ значение ↔ состояние ↔ событие ↔ сигнал ↔ приоритет ↔ действие ↔ критерий
↔ оценка ↔ принятие решений

Для полноты представления необходимо ввести актуализацию понятий в процессе компьютеринга (вычислений, вывода, преобразований, навигации, ...).

В этом случае естественным образом получаются различные *стратегии компьютеринга*.

Любой информационный артефакт для количественной и качественной оценки должен быть проинтерпретирован в рамках ИТ-системы (системы измерений длина-время). Так можно получить аналоги в информационной области законов сохранения энергии и мощности [3, 5].

В рамках интерпретации получают оценки сложности структуры и функции созданного информационного артефакта. В *динамической сложности* имеют смысл комбинации:

L — абстрактный размер памяти или размер вычислительного поля для параллельных вычислений (расстояние доступа к информации);

Таблица 2. Матрица стратегий компьютеринга

| Исполнение | Безусловное | | Условное | | Смешанное | |
|------------------|-------------|-------------|------------|-------------|------------|-------------|
| | Синхронное | Асинхронное | Синхронное | Асинхронное | Синхронное | Асинхронное |
| Последовательное | | | | | | |
| Параллельное | | | | | | |
| Конкурентное | | | | | | |

L^3 — полный объём памяти или объём процессорного поля при неограниченном параллелизме;

L^3T — аренда памяти для последовательного вычисления или вычислительная работа для параллельного вычисления;

T — время для последовательного вычисления или задержка для параллельного вычисления.

В статической сложности можно рассмотреть комбинации:

L^3 — объём алгоритма (в элементах структуры);

L — диаметр алгоритма (размер в 3-мерной памяти);

L^2 — отношение объёма к диаметру — относительная поверхность алгоритма (площадь его сечения — степень «запутанности»).



Рис. 4. Процесс разрешения противоречий на основе организаторов степеней свобод

Вопросы о законах сохранения различных вычисленных таким образом сложностных параметров задач гораздо более сложны, чем соответствующие вопросы в физике. Они соответствуют различным trade-off-проблемам теории сложности. Например, не ясно, можно ли сокращать объём вычислительной работы за счёт увеличения задержки (которая первоначально была существенно, но не катастрофически меньше, чем этот объём при последовательном вычислении).

Сложностная картина способна дать относительно полное представление для формирования конструктивных механизмов, в рамках которых можно выделить два основных направления, обусловленных симбиозом.

Первое направление это решение творческих задач на основе либернетического и генетического представления систем (см. рис. 4).

Второе направление, наряду с логическим подходом, позволяет решать задачи параметрического, структурного и базового уровня описания системы (см. табл. 3).

Таблица 3. Уровни управления и синтеза

| Уровни управления | Редукторы степеней свобод | Генераторы степеней свобод | Примечание |
|--|--|---|---|
| Параметрический | $x_i \uparrow \rightarrow \text{const}$ | $\text{const} \rightarrow x_i \uparrow$ | Изменение числа степеней свобод, фиксации, регуляции |
| | $x_i \uparrow \rightarrow x_k \downarrow, x_i \downarrow \rightarrow x_k \uparrow$ – отрицательная | | |
| | $x_i \downarrow \rightarrow x_k \downarrow, x_i \uparrow \rightarrow x_k \uparrow$ – положительная | | |
| Структурный e_q – элементы | $e_i, e_j \rightarrow e_k$ – «склейка» | $e_k \rightarrow e_i, e_j$ – «рубка» | Изменение структуры |
| | | $\pm(e_i)$ | Нет однозначности |
| r_v – связи | $+(r_m)$ | $-(r_n)$ | Изменение структуры, синергии и инварианты |
| Базисный – статический – динамический | $x_i \rightarrow y_k$ – изменение базиса переменных | | изменение субстанции или точки зрения, изменение функционального насыщения (преобразования) |
| | $x_i \in [a, b]_i \rightarrow x_i \in [c, d]_i$ – изменение интервалов существования переменных | | |
| | $u_i = T_{ip} \bar{u}_p, \bar{u}_i \in O x_i, u_p \in A z_p$ | | |
| | $u_i \xrightarrow{F} u_k \Rightarrow u_i \xrightarrow{G} u_k$ | | |

Таким образом, изложенная в статье концепция описания ИТ-сферы,

во-первых, интегрирует «хорошо продифференцированные» области знаний и выделенные роли в ИТ-сфере, при этом возникает целостное представление о процессах в рамках жизненных циклов человека, цивилизации, информационных систем;

во-вторых, порождает нетрадиционные исследования в оценке ИТ-сферы;

в-третьих, является мощным катализатором устойчивого развития человека, общества и окружающей природы.

СПИСОК ЛИТЕРАТУРЫ

1. Бельтюков А. П., Маслов С. Г., Морозов О. А. Либернетическая парадигма в ИТ-сфере // Технологии информатизации профессиональной деятельности (в науке, образовании и промышленности): Сб. тр. II Всерос. науч. конф. с междунар. участием. Ч. I. — Ижевск: ООО Информационно-издательский центр «Бон Анца», 2008. – С. 37–52.

2. Вольфенгаген В. Э. Конструкции языков программирования. Приёмы описания — М.: АО «Центр ЮрИнфоР» 2001. — 276 с.
3. Кузнецов О. Л., Большаков Б. Е. Устойчивое развитие: Научные основы проектирования в системе природа-общество-человек: Учебник. Санкт-Петербург; Москва; Дубна: Гуманистика, 2001. — 616 с.
4. Непейвода Н. Н., Скопин И. Н. Основания программирования. — Москва; Ижевск: Институт компьютерных исследований, 2003. — 668 с.
5. Петров А. Е. Тензорный метод двойственных сетей. — М.: ООО «Центр информационных технологий в природопользовании», 2007. — 496 с.

Поступила в редакцию 30.03.09

A. P. Beltukov, S. G. Maslov, O. A. Morozov
Constructive activities of IT-sphere

Problems and conditions connected with the forming of the constructive mechanisms of IT-spheres are discussed.

Keywords: IT-sphere, system descriptions, lingual form, images and means, constructive activity, informational artifact, programming, physical reality, LT-system, libernetics, strategies of computing.

Mathematical Subject Classifications:

Бельтюков Анатолий Петрович,
д. ф.-м. н., профессор кафедры теоретических основ информатики,
Удмуртский государственный университет,
426034, Россия, г. Ижевск, ул. Университетская, 1 (корп. 6)
E-mail: belt@udsu.ru

Маслов Сергей Геннадьевич,
к. т. н., доцент кафедры теоретических основ информатики,
Удмуртский государственный университет,
426034, Россия, г. Ижевск, ул. Университетская, 1 (корп. 6)
E-mail: mshsci@yahoo.ru

Морозов Олег Анатольевич,
начальник отдела компьютерных сетей университета,
Удмуртский государственный университет,
426034, Россия, г. Ижевск, ул. Университетская, 1 (корп. 6)
E-mail: oam_oam@gmail.ru