

УДК 681.3.06

И.В.Редько

ДЕСКРИПТИВНЫЕ СРЕДЫ: ИНТЕГРАЦИОННЫЕ ОСНОВАНИЯ

Данная статья посвящена рассмотрению интеграционных аспектов дескрипирования. Дается развернутая экспликация основных понятий дескриптологии: дескрипции, дескриптивной среды, действия, полиады, презентативной акции, композиции. Исследуются аспекты генезиса сущностей в дескриптивной среде.

Дескриптивные среды (ДС) – это среды, которые поддерживают дескрипирование сущностей (объектов, процессов, явлений и др.) При этом дескрипирование сущности понимается как процесс построения ее дескрипции (описания). В рамках предлагаемой прагматики *сущность* – парадигмная категория, т.е. основное и единственное понятие, не сводимое ни к каким другим. Трактуются это понятие в самом широком смысле, включая всевозможные реальные и идеальные, конкретные и абстрактные, статические и динамические, частные и общие сущности, в том числе сущности типа свойств сущностей, свойств высшего типа и даже *универсум сущностей* как бытие сущего. Среда программирования (СП), включающие, в частности, среды интеграции (СИ) процессов программирования и решения задач – репрезентативные примеры ДС, в которых дескрипирование суть процессы построения дескрипций типа программ и решений. *Интеграционная парадигма* в процессах информатизации, в частности программировании, да и не только в нем, занимает основополагающее место (см. [1-4] и библиографии к ним). Не являются здесь исключением и дескриптивные среды. Настоящая статья посвящена созданию оснований для дальнейшего развития этой парадигмы применительно к ДС. С этой целью посмотрим с общей точки зрения на *сущностную платформу* как основу дескрипирования. При этом все неопределяемые понятия и используемые здесь результаты понимаются в смысле [1, 4].

Сущностная платформа в первом приближении. Объем понятия сущности предельно велик, а значит, малосодержателен. Поэтому данное понятие необходимо обо-

гатить. Осуществим это в виде постепенных обогащений, каждое из которых удовлетворяет принципу достаточных оснований Лейбница ([5], стр. 212, 234, 418, 564). Применительно к цели работы сформулируем его так: *обогащение должно быть достаточно содержательным и не быть обремененным спецификой*. Такая формулировка, как и другие неформальные специализации принципа Лейбница, носит методологический, а не логико-математический характер. Тем не менее она позволяет обогащение содержания понятия сущности свести к пошаговой прагматико-мотивированной типизации универсума сущностей *T*. Условимся называть ее *типизацией, обусловленной прагматикой* (ТОП).

Эпистемологической основой ТОП является тот факт, что в познавательном смысле важны не столько *сущности сами по себе* (т.е. сущности вне познания), сколько их проявления. При этом *проявления* понимаются в широком смысле, включающем как всевозможные процессы проявления, так и их абстракции, т.е. результаты этих процессов (собственно проявления). В качестве примеров таких проявлений могут выступать процессы восхождения и нисхождения от абстрактного к конкретному и, что особенно важно, процессы, поддерживающие смешанные (комбинированные) стратегии рассмотрения сущностей. Важность последних состоит в том, что они не обременены абсолютизацией ни нисхождения, ни восхождения. Издержки такой обременительности восхождения по Гегелю хорошо выразил один из основоположников математической логики Ч.С. Пирс: «План Гегеля, сводившийся к разворачиванию всего из абстрактнейшего понятия посредством диалектической процедуры, – хотя и не так абсур-

ден, как думают эксперименталисты, а наоборот, представляет собой одну из необходимых частей научного дискурса, – упускает из виду слабость индивидуального человека, ибо последнему удержать в руках подобное оружие попросту не достаёт сил». [6, стр. 72-73]. Само собой разумеется, что абсолютизация дуальной стратегии – нисхождения не лучший способ оптимизации издержек. Как примеры проявлений могут выступать также статические и динамические модели сущностей, описания, в частности определения этих моделей, в естественных и искусственных языках. Непосредственно из неформального определения понятия проявления вытекает, что оно не тождественно понятию свойства¹. Проявления – это не просто свойства типа тех или иных черт сущности, но и сущностей во множественном числе (не путать с сущностью типа множества!). В этом смысле понятие проявления является более общим, по сравнению с понятием свойства. Оно не обременено спецификой единичности, что позволяет корректно говорить, в частности, о сущностях, заданных не актуально, а потенциально. Это является принципиально значимым в природе проявления. Поэтому все дальнейшие необходимые обогащения этого понятия должны быть такими, которые сохраняют данное его свойство. Последнее лежит в основе прагматико-мотивированных описаний проявлений в самом общем смысле, включающих как описания внутренних, так и внешних свойств сущностей. Прагматическую базу всех этих возможностей составляют не просто сущности, а сущности высшего порядка – среды их существования (Среда существования понимается здесь как первичное понятие, обогащаемое в рамках последующих шагов ТОП.) Условимся такие среды называть *дескрипциями*. (Дескрипция как среда существования сущностей несет в

¹ Содержательное отличие понятий проявления и свойства состоит в том, что свойство это не самодостаточное понятие. Оно всегда связано с некоторой единичной сущностью, которая естественно должна быть так или иначе изначально дана. Проявление же есть самодостаточной сущностью как процесс или результат процесса (его абстракция).

себе в частности свойство «возможность корректно говорить о факте существования сущностей среды».)

Дескрипция – центральное понятие сущностной платформы. Содержание его адекватно отражает глубинную природу современных представлений об описаниях сущностей. При этом оно не ограничивается поверхностными проявлениями, тем более синтаксическими, как это чаще всего бывает, а вскрывает их истинную природу. Потому дескрипция вполне может рассматриваться как универсальное средство познания сущностей, воплощаемое, например, в форме тех или иных прагматико-обусловленных языков. Ввиду этого сущности типа дескрипций индивидуализируются (выделяются) в универсуме T . Сказанное выше даёт основания рассматривать эту индивидуализацию в качестве первого шага ТОП. Последующие обогащения сущностной платформы сводятся к обогащению T на основе понятия дескрипции. С этой целью предварительно уточним в соответствии с принципом Лейбница само это понятие, эксплицировав в рамках ТОП среды существования сущностей (ССС).

Под *средой существования сущностей* будем понимать сущность, которая имеет как *составляющие* типа сущностей, так и *окружение*, в котором они проявляются. Содержательно окружение можно представлять в виде сущности, поддерживающей взаимозависимость сущностей в среде их существования. При этом важно сразу подчеркнуть, что окружение не отождествляется с СССР. Разные СССР могут иметь одно и то же окружение. Связано это с тем, что разные составы сущностей могут проявляться одинаково. Особый интерес представляют окружения СССР, инвариантные относительно взаимно однозначной замены их сущностей, в частности перестановки, любыми другими сущностями из T . Договоримся такие окружения именовать *компаундами*.

Непосредственно из определения понятия компаунда вытекает, что они представляют собой такие окружения СССР, которые, содержательно говоря, независимы от составляющих, т.е. совместимы с любыми другими

составляющими. Указанное свойство позволяет естественно представлять компаунды в форме схем, по отношению к которым сами компаунды выступают их содержанием. Принципиально здесь то, что в таком представлении доминирует содержание, а не форма. В случае языковых дескрипций это соответствует семантико-синтаксическому подходу, в противоположность традиционному их синтактико-семантическому представлению, что в большей степени согласуется с принципом подчиненности.

Важность понятия компаунда вытекает из следующего принципа.

Принцип компаундности. Дескрипции экспликативно сводим к компаундам.

Этот принцип является естественным обобщением одноименного принципа, изложенного в [4]. Из него и принципа дескриптивности вытекает, что сущности экспликативно сводимы к компаундам. Это дает основания рассматривать сущности типа компаундов как важное дальнейшее обогащение сущностной платформы. Поэтому индивидуализация в универсуме T сущностей типа компаундов рассматривается нами как второй шаг ТОП. Последующие шаги поддерживаются более конкретными прагматиками по сравнению с предыдущими. В качестве «общего знаменателя» этих прагматик выступает интеграционное начало дескриптивных сред. Вскрытие природы последнего естественно осуществить на базе интеграционной платформы дескриптивирования.

Интеграционная платформа. В самом общем смысле интеграционная платформа (ИП) – это платформа, поддерживающая взаимозависимости частей сущностей в рамках целого. При этом сами эти части могут присутствовать в нем как актуально, так и потенциально. Логико-математический фундамент этих возможностей составляют полиады.

Понятие полиады задается следующим индуктивным определением:

1. Абстрактные сущности суть полиады.
2. Любые множества, кортежи и импульсы полиад суть полиады.
3. Других полиад нет.

Их значимость является следствием следующего принципа.

Принцип полиадности. Компаунды экспликативно сводимы к полиадам.

Основополагающее значение данного принципа в интеграционной проблематике освещено в [3, 4]. Поэтому здесь мы отметим лишь те аспекты, которые не нашли там должного отражения. С этой целью рассмотрим пример задания на разработку целостной системы информатизации, составляющими которой являются подсистемы, вырабатывающие значения параметров, обрабатываемые управляющей системой. Относительно подсистем известно лишь, что это процессы, уточняемые во время разработки, опытной эксплуатации, сопровождения и даже промышленной эксплуатации. Причем одним из основных требований здесь является сохранение инвестиций. Отсюда непосредственно следует, что управляющую систему не нужно жестко связывать как с актуальностью отдельных значений, вырабатываемых подсистемами, так и со структурой, которую они в целом образуют. Ведь в этом случае фактически игнорируется важнейшая специфика подсистем, в особенности их процессональность. При этом нужно иметь в виду, что поддержка последней «в розницу» входит в противоречие с требованием сохранения инвестиций. Поддержать же процессональность «оптом» – значит обеспечить гибкую настройку системы на условия, интегрирующие в себе прагматико-мотивированную поддержку частных подсистем в рамках общей системы управления. Однако нетрудно понять, что это в принципе невозможно сделать в рамках традиционных подходов. Связано это с необходимостью поддержания свойства потенциальности (модальности) подсистем – следствия их процессональности.

Чтобы разобраться в этом неформальном анализе более обстоятельно, конкретизируем задачу. С этой целью в стиле объектно-ориентированного подхода посмотрим прежде всего на подсистемы с абстрактно-инкапсулятивной точки зрения. А именно вовлечем в рассмотрение лишь их абстрактное свойство – «быть импульсом (абстрактным про-

цессом) $i, i \in \overline{1, n}$, вырабатывающим значение переменной x_i ». Все другие свойства этих подсистем инкапсулируются. В соответствии с принятым подходом разработка целостной системы информатизации сводится к разработке управляющей системы, семантика которой задается функцией $f(x_1, \dots, x_n)$ с переменными x_1, \dots, x_n , значения которых вырабатываются упомянутыми импульсами (подсистемами). Чтобы осуществление этого произошло в соответствии с требованием сохранения инвестиций, необходимо связывать разработку управляющей системы не непосредственно с самой функцией $f(x_1, \dots, x_n)$, а с ее математической моделью как абстракцией от всей «рознично-обременительной» специфики функции $f(x_1, \dots, x_n)$. Конечно, проблема выбора типа модели не может быть решена однозначно, а определяется лишь, во-первых, степенью продвинутости наших представлений о предмете и средствах моделирования и, во-вторых, прагматикой самого моделирования. Как указывалось в [4], можно выделить по крайней мере три типа таких моделей: классическую, опирающуюся на традиционную теоретико-множественную платформу, неоклассическую, основу которой составляет триада именного множества, именной функции и композиции и неклассическую, в основании которой лежит концепция дескриптивных сред.

Уже использование классической модели функции обеспечивает серьезное продвижение на пути создания упомянутых систем управления. Тем более сказанное относится к неоклассическим моделям функций, которые в области разработки и внедрения программных систем прошли солидную проверку временем. Однако серьезный анализ всех трех подходов, проведенный в [4], думается, мотивировано показал, что поставленная в начале этого раздела задача может быть адекватно ее прагматике решена только на основе неклассических моделей. Чтобы еще раз акцентировать внимание на истинной причине указанного обстоятельства, целесообразно на последний (неклассический) под-

ход посмотреть с позиции второго (неоклассического) подхода. В общих чертах это рассмотрение можно резюмировать следующим образом. Именные множества обобщаются до *импульсных множеств* – множеств импульсов, именные функции – до импульсных функций (*акций*), изначально заданные неоклассические композиции, в том числе прагматически полная их совокупность, – до инициально порождаемых классов неклассических композиций (*презентативных акций*).

Построение интеграционных оснований дескриптивных сред было направлено главным образом на обогащение неоклассической композиционной парадигмы до неклассической. В этом обогащении аккумулируются все основные особенности такого построения в рамках ТОП, детально рассмотренные в [1-4]. Здесь хотелось бы лишь подчеркнуть, что многие из этих особенностей проявляются уже на предыдущих шагах типизации универсума сущностей T , например переходе от именных до импульсных функций.

Пусть $f = f(x_1, \dots, x_n)$ – функция, трактуемая как именная, т.е. сопоставляющая именованным множествам $\{(x_1, a_1), \dots, (x_n, a_n)\}$, где $x_i, i \in \overline{1, n}$ – имена (переменные), а a_i – соответствующие им денотаты (значения), значения функции f , заданные теми или иными правилами (законами). Такая трактовка, как хорошо известно, существенным образом расширяет интеграционные возможности традиционных пониманий функций. Ведь понятие n -арной функции, составляющее основу таких пониманий представляет собой всего лишь частный и, к сожалению, очень обременительный в интеграционном плане вид концептуально единого рода именных функций. Это именные функции со стандартными именами $1, \dots, n$ и обычным порядком (компаундом), заданным на них и являющимся «прокрустовым ложем» в решениях реальных задач. Однако, невзирая на упомянутое расширение возможностей, формализовать семантику управляющей системы интеграционной среды информатизации, разрабатывае-

мой в рамках указанных выше требований, в виде именной функции f , также обременительно. Дело в том, что правила, задающие вычисления значений этой функции на именных множествах вида $\{(x_1, a_1), \dots, (x_n, a_n)\}$, можно использовать лишь тогда, когда все пары (элементы) (x_i, a_i) , $i = \overline{1, n}$, этого именного множества являются актуальными сущностями (данностями), т.е. изначально заданы. Но это противоречит требованиям на разработку целостной системы информатизации. Согласно им это неданности (потенциальные сущности), формируемые подсистемами, поддерживаемыми процессами с предпосылками $x_i, i = \overline{1, n}$, и последствиями, если они имеются, из области потенциально возможных значений (денотатов) x_i , интерпретируемых как переменные.

Из вышесказанного вытекает, что формализация семантики управляющей системы и тем более всей ее интеграционной среды сопряжена как минимум с прагматикомотивированной экспликацией интуитивного понятия процесса. Однако, как показано в [4], этого не достаточно. Формализация на этой основе обременена общностью (!). Издержки этой обременительности – прямое следствие абсолютизации упомянутой выше стратегии восхождения. Они столь же нежелательны, как и издержки излишней конкретизации². Поэтому объем общего понятия процесса по необходимости следует ограничить абстрактными процессами типа действий. Иными словами, осуществить по отношению к процессам неформальную операцию *введения абстракции*, инкапсулировав при этом излишнюю их конкретность, связанную с пошаговостью. Индивидуализация этого вида абстракции в рамках ТОП существенным образом способствует адекватному решению исходной задачи. Но ее также недостаточно, так как и в рамках указанной индивидуализации обременительной остается специфика действий. Обременительность проявляется в принци-

альной невозможности адекватной поддержки иерархии подсистем в рамках целостной интеграционной среды [7]. Связано это с тем, что такая поддержка сопряжена с проблемой экспликации суперпозиции подходящих функций. Функции, заданные на множествах действий, здесь не являются таковыми, так как последние тоже излишне специфичны. Поэтому необходим более высокий тип абстракции. Конкретнее говоря, следует вовлечь в рассмотрение лишь «минимальное» проявление действий типа компаунда с единой предпосылкой и возможным последствием. Все другие проявления до поры до времени инкапсулируются, чтобы их вскрыть в нужное время и в нужном месте. В связи с этим индивидуализируем в универсуме T сущности типа упомянутых компаундов и условимся называть их *импульсами*. При этом заметим, что такой переход поддерживается операцией введения абстракции высшего типа.

С содержательной точки зрения понятие импульса является естественным обобщением понятия пары, при котором вовлекается в рассмотрение лишь возможность иметь второй компонент, однако не требуется обязательная актуализация этой возможности. Это позволяет наполнить принципиально новым содержанием понятие именного множества, обобщив его до *импульсного множества*. Наконец, это делает возможным решение главной проблемы – экспликации понятия *импульсной функции* как функции, заданной на импульсных множествах. Значимость этих функций вытекает из их адекватности упомянутой интеграционной проблематике. Ведь сохраняя высокую содержательность именных, тем более n -арных функций, данное понятие очевидно освобождено от обременительной специфики последних.

Проиллюстрируем это на простейшем примере. Пусть функция $f = f(x, y) = x + y$, где x и y – переменные, принимающие, для определенности, натуральные значения. Классический вариант экспликации этой функции – бинарная операция, сопоставляющая парам $\langle m, n \rangle$, где m и n – натуральные числа, число $m + n$. Как видим, такое уточне-

² К сожалению, этот аспект не только сплошь и рядом упускается из виду, но к тому же часто возводится в ранг непререкаемой догмы рассмотрения.

ние не является адекватным в рамках предлагаемой прагматики уже хотя бы потому, что здесь за рамки объективных рассмотрений вынесены сущности типа переменных x, y , которые тем не менее объективно используются в интуитивном вскрытии природы функции. Неоклассический вариант – именная функция, которая ставит в соответствие именованным множествам вида $\{(x, m), (y, n)\}$, где x, y, m и n имеют прежний смысл, натуральное число $m+n$. Данное уточнение явным образом вовлекает в рассмотрение переменные x и y в качестве имен денотатов m и n . Адекватность такого вовлечения проявляется, в первую очередь, в задачах, связанных с использованием вычислительной техники. Ведь именно в этой прагматике, мы не можем себе позволить отнести что-либо существенное из природы эксплицируемой сущности, в данном случае функции f , к области субъективного рассмотрения, как это сделано в случае классической экспликации. Наконец, неклассический случай экспликации исходной функции – это импульсная функция, которая импульсным множествам вида $\{i_x, i_y\}$, где x и y имеют прежний смысл, а i_x и i_y представляют собой импульсы, предпосылками которых являются соответственно переменные x и y , ставит в соответствие значения, представляющие собой суммы возможных последствий упомянутых импульсов. Данный вариант уточнения является, очевидно, более предпочтительным в смысле адекватности решаемой задаче – экспликации функции f по сравнению даже с неоклассическим и тем более с классическим вариантами. Содержательно говоря, неадекватность «именной» экспликации выражается в определенной подмене понятий. Здесь уточнение самой сущности типа функции как правила $f(x, y) = x + y$ заменяется уточнением возможных функциональных ситуаций. Проявляется это в необходимости вовлечения в определение излишней конкретики, которой, кстати, нет в исходном интуитивном описании функции – непосредственно

заданных значений переменных как денотатов имен x и y . Неклассическая же экспликация благодаря *косвенному* заданию значений переменных как последствий импульсов, очевидно, лишена этой обременительной специфики.

Конечно, адекватность импульсных функций, продемонстрированная на простейшем примере, важна сама по себе. Однако намного важнее, что на их основе можно эксплицировать синтез сколь угодно сложных функций из сравнительно более простых. В связи с этим снова рассмотрим простейший пример. Пусть требуется синтезировать импульсную функцию $f = f(x, y, z) = x + y \cdot z$ из более простых. Интуитивно ясно, что функция f есть суперпозицией функции умножения $f_1 = f_1(y, z) = y \cdot z$ в функцию сложения $f_2 = f_2(x, u) = x + u$. Поэтому задача эксплицировать требуемый синтез сводится к экспликации операции суперпозиции в классе импульсных функций. Такая экспликация была дана в [4]. Применительно к рассматриваемому случаю $f = S^u(f_2, f_1)$. Очевидно, что данное уточнение совершенно адекватно интуитивным представлениям о естественном решении проблемы синтеза функции f . В отличие от неклассического подхода проблема синтеза даже в этом простейшем случае, в рамках неоклассического и тем более классического подходов, не только не может быть решена, но даже, как следует из вышесказанного, адекватно поставлена.

Заключение

Обсуждаемые в статье проблемы интеграционных сред возникли в процессе разработки нами систем различной сложности. Особенно они проявились при создании программных комплексов типа СКИФ (Система Комплексной Информатизации), Refernet как ядра сетевых технологий и экспликативных CASE-технологий, которые в рамках единой компьютерно поддерживаемой технической дисциплины способствовали бы созданию интеграционных сред. Главным отличительным свойством этих проблем является то, что трудности их решений в большой мере про-

являются даже на микроуровне, не говоря уже о макроуровне. Иллюстрация их на простейших примерах представляется убедительным подтверждением этого.

1. *Редько В.Н.* Основания дескриптологии // Кибернетика и системный анализ. – 2003. – № 5. – С.16 – 36.
2. *Редько В.Н., Редько И.В., Гришко Н.В.* Дескриптологическая среда информационных технологий // Проблемы программирования. – 2004. – №2. – С. 65-73.
3. *Редько В.Н., Редько И.В.* Дескриптивные системы: ретроспективы и перспективы // Вісник Київського національного університету ім. Т.Шевченка. Сер. Фіз.-мат.науки. Спец. вып.– 2004. – С.68-75.
4. *Редько И.В.* Прагматические основания дескриптивных сред // Проблемы программирования. – 2005. – №3. – С. 3-25
5. *Лейбниц Г. В.* Сочинения: В 4-х т. – М.: Изд-во «Мысль», 1982. – Т.1. – 636 с.
6. *Пирс Ч.С.* Принципы философии. – СПб.: Санкт-Петербургское философское общество, 2001. – Т.2. – 320 с.

7. *Редько И.В.* Экспликативный базис интеграционных сред // Проблемы программирования. – 2004. – №3. – С. 59-65.

Получено 05.01.05

Об авторе

Редько Игорь Владимирович
канд. физ.-мат. наук, доцент

Место работы

Национальный технический университет
Украины «Киевский политехнический
институт»
03056, Киев, проспект Победы, 37
факультет электроники, корп. 12
Тел.: 241 8039
E-mail: sms@ipnet.kiev.ua