

## РЕЗЕРВЫ ПРОГРАММИРОВАНИЯ ДЕЯТЕЛЬНОСТИ. 3-МЕРНАЯ ПРОГРАММНАЯ МЕТАМОДЕЛЬ СИСТЕМЫ ДЕЯТЕЛЬНОСТИ

---

**Анотація.** Програмування є одним із видів моделювання діяльності. Існують такі можливості підвищення ефективності програмування діяльності: 1) ревізія і уточнення понятійно-термінологічного апарату в даній предметній області; 2) розширення можливостей створення і використання програмних моделей діяльності за рахунок застосування спеціальних систем; 3) створення та використання  $n$ -мірних програмних метамodelей діяльності. Стаття присвячена розгляду останньої із зазначених можливостей.

**Ключові слова:** діяльність, дія, модель, метамодель, процес, операція, операнд, оператор, спосіб виконання операції, програма, програмування діяльності.

**Аннотация.** Программирование является одним из видов моделирования деятельности. Имеются следующие возможности повышения эффективности программирования деятельности: 1) ревизия и уточнение понятийно-терминологического аппарата в рассматриваемой предметной области; 2) расширение возможностей создания и использования программных моделей деятельности за счет применения специальных систем; 3) создание и использование  $n$ -мерных программных метамodelей деятельности. Статья посвящена рассмотрению последней из указанных возможностей.

**Ключевые слова:** деятельность, действие, модель, метамодель, процесс, операция, операнд, оператор, способ выполнения операции, программа, программирование деятельности.

**Abstract.** Programming is one of types of simulation an activity. There are the followings possibilities to increase the efficiency of activity programming: 1) revision and clarification of concepts and terminology in the examined subject domain; 2) expansion of possibilities of creation and use of programmatic activity models due to application of the special systems; 3) creation and use of the  $n$ -dimensional programmatic activity metamodels. The article is devoted to consideration of the last of indicated possibilities.

**Key words:** activity, action, model, metamodel, process, operation, operand, operator, method of implementation of operation, program, activity programming.

### 1. Введение

Программирование деятельности – понятие, которое впервые было введено в отечественный научно-технологический обиход в работе [1] и которое, как нам представляется, адекватно отражает суть моделирования аспектов человеческой деятельности, касающихся ее процедурной регламентации, в отличие от понятия «моделирование бизнес-процессов».

В статье [2] были определены следующие направления, работая по которым можно увеличить практическую отдачу от программирования деятельности:

совершенствование понятийно-терминологического аппарата в области создания и использования высокоуровневых программ, регламентирующих осуществление технологических процессов в самых разных сферах человеческой деятельности;

поддержка создания и использования программ деятельности соответствующими инструментальными средствами;

исследование, создание и использование типовых  $n$ -мерных программных метамodelей систем деятельности.

Два первых направления были рассмотрены соответственно в [2, 3]. Данная статья посвящена рассмотрению последнего из указанных направлений.

## 2. Понятие $n$ -мерной программной метамодели системы деятельности

Для разъяснения того, что понимается под « $n$ -мерной программной метамоделью системы деятельности», нам потребуется объяснить его генезис – ввести и обсудить ряд вспомогательных понятий, а именно:

уровень абстрагирования/конкретизации, присущий идентифицированному типу объекта или идентифицированной операции;

метамодель программы;

$n$ -мерная система деятельности;

$n$ -мерная система программ деятельности;

$n$ -мерная метамодель системы деятельности.

Когда мы говорим, например, «Изготовить изделие (Изготовление изделия) по имеющемуся чертежу», мы совершенно четко идентифицируем конкретную специфику вида деятельности, который имеем в виду. Собственно, эти слова – уже оператор, ссылающийся на конкретную операцию, однако операцию, введенную в рассмотрение на достаточно высоком уровне абстракции. То есть, уже даже по предложенной формулировке мы четко понимаем, о чем идет (может идти) речь. И, наверное, чувствуем, что можно конкретизировать определение этой операции, во-первых, за счет того, какова природа изделия<sup>1</sup>, как оформлен чертеж<sup>2</sup>, на кого мы рассчитываем как на исполнителя и т.д. А это уже конкретизация входа/выхода операции и ее преобразователя.

Таким образом, мы можем говорить о том, что в общем случае определение операции, сделанное на некотором уровне абстракции, может быть и (еще больше) конкретизировано (понижение уровня абстракции), и (еще больше) обобщено (повышение уровня абстракции).

Представление об уровнях абстракции описаний, которое мы применили к описаниям типов объектов и операций, можно распространить и на программы, При этом взаимоотношения метамодели программы и собственно программы<sup>3</sup>, занимающих два соседние уровня абстракции, можно охарактеризовать следующим образом. Метамоделью программы будем называть программу, которая:

представлена в некоторой определенной нотации;

является описанием способа осуществления определенного вида деятельности (выполнения операции) по определенной схеме [4];

предназначена не для непосредственного выполнения, а для использования в качестве прототипа для построения метамodelей более низкого уровня абстракции вплоть до «выполняемых» программ с привлечением для их описания языков программирования.

При переходе от метамодели к модели (выполняемой программе) можно допустить<sup>4</sup> некоторое развитие и/или изменение метамодели.

В предыдущих статьях [2, 3] мы особо останавливались на таком понятии, как процесс осуществления деятельности, и на значении такого информационного объекта, как описание (протокол) процесса осуществления деятельности, который сам по себе может быть объектом как моделирования, так и метамоделирования.

А теперь представим себе два вида деятельности, назначение одного из которых заключается в некотором влиянии на другой. В терминах процессов как реальных объектов это означает, что выход одного процесса является компонентом другого. Такие процессы

<sup>1</sup> Например, мы можем воспользоваться классификатором видов продукции.

<sup>2</sup> Например, мы можем указать стандарт(ы), в соответствии с которым(и) чертеж должен быть оформлен.

<sup>3</sup> Здесь мы исходим из того, что программа является моделью способа осуществления деятельности, которая может иметь метамодель.

<sup>4</sup> Разумеется, при наличии обоснований.

являются связанными между собой через общие объекты. Процесс (Б), влияющий на другой процесс (А), назовем фактор-процессом по отношению к А. Процессы А и Б можно образно назвать «ортогональными» друг к другу в некоторой системе координат, оси которой соответствуют назначениям процессов<sup>5</sup>. В общем случае процесс А может иметь другие фактор-процессы, отличные от Б, который, в свою очередь, может иметь собственные фактор-процессы и т.д., и в определенной предметной области можно усмотреть наличие  $n$ -связанных между собой видов деятельности, реализующихся через связанные процессы, рассматривая их как  $n$ -мерную систему связанной деятельности<sup>6</sup>.

Пусть теперь процесс Б осуществляется в соответствии с программой, определяющей способ осуществления деятельности. Ясно, что необходимым условием для такой программы являются протоколирование деятельности, реализуемой процессом А, и доступ к этому протоколу. При этом, если способ осуществления деятельности А также задан опорной программой, протокол процесса А будет продуктом выполнения этой программы так, как это описано в [3]. Если же для каждого вида деятельности, входящего в  $n$ -мерную систему связанной деятельности, имеется своя программа, задающая способ осуществления этого вида деятельности, то можно говорить о наличии у этой системы деятельности  $n$ -мерной системы программ.

Приостанавливая на этом развитие нашего примера, выдвинем 1-ю гипотезу о возможности существования практически целесообразных программных систем, в которых можно усмотреть присутствие  $n$ -мерности в предложенном выше смысле, т.е.  $n$ -мерных систем программ деятельности.

Наконец, предположим, что мы уже имеем некоторую  $n$ -мерную систему программ деятельности, отталкиваясь от которой строим ее общую метамодель заменой ее программных компонентов содержательными метамоделями программ. И в итоге получаем  $n$ -мерную программную метамодель. Соответственно, выдвинем 2-ю гипотезу о возможности существования практически целесообразных  $n$ -мерных метамodelей систем деятельности.

Как нам представляется, цель данной статьи будет достигнута, если нам удастся доказать правомерность обеих выдвинутых нами гипотез. Свои доказательства мы намерены осуществлять построением содержательных примеров программ. При этом следует сделать одну оговорку: в данных примерах нам, по-видимому, придется все время оставаться на уровне метамodelей программ. Но мы надеемся, что когда речь будет идти о возможности/целесообразности перехода от метамodelей к программам, эта возможность/целесообразность окажется очевидной.

### 3. Нотации и оформление примеров

Определим правила оформления наших примеров.

Для записи примеров программ нам достаточно воспользоваться некоторой нестрогой нотацией, а именно, программу мы будем представлять в виде трех компонентов:

заголовок программы – оператор, состоящий из имени (кода) операции и списка имен параметров, возможно, пустого и заключенного в скобки;

группа, возможно, пустая, операторов, декларирующих тип и имя (код) обрабатываемых и/или используемых в программе объектов, а также создающих экземпляры объектов;

---

<sup>5</sup> Разумеется, мы используем здесь термины «ортогональность», «система координат», «координатная ось» отнюдь не в их обычном математическом смысле, а как метафоры, надеясь когда-нибудь впоследствии найти более подходящие названия для вводимых в рассмотрение образов.

<sup>6</sup> В силу предложенной метафоры «назначение вида деятельности» = «самостоятельное измерение для системы деятельности».

тело программы – графическое описание порядка выполнения вспомогательных операций. Символы заимствованы из нотации BPMN [5], и их значение представлено в табл. 1.

Таблица 1. Графические символы и их значения

Символ	Значение символа
	Начало (старт) программы
	Промежуточное событие
	Окончание (финиш) программы
	Операционный блок. Символ позволяет расположить в нем текст, который в простейшем случае представляет собой оператор, хотя нет оснований запрещать в нем размещение и последовательности операторов
	Операционный блок, вырабатывающий значение условия. Символ позволяет расположить в нем текст, который в простейшем случае представляет собой оператор, хотя нет оснований запрещать в нем размещение и последовательности операторов
	Так называемая «плавательная дорожка». В нашем случае будет использована для показа продолжающегося во времени процесса, с которым взаимодействует программа
	Передача управления от одного символа к другому. Символ, при необходимости, позволяет ассоциировать с ним текст, который дополнительно описывает характеристику перехода
	Условная передача управления от одного символа к другому. Символ, при необходимости, позволяет ассоциировать с ним текст, который дополнительно описывает характеристику (условие) перехода
	Передача управления от одного символа к другому по умолчанию. Символ, при необходимости, позволяет ассоциировать с ним текст, который дополнительно описывает характеристику (условие) перехода

Каждый новый пример, как правило, связан с введением в рассмотрение новых типов объектов и/или операций, которые последовательно, по мере продвижения от примера к примеру, регистрируются в двух группах таблиц соответственно:

группа таблиц «2» (табл. 2.1, 2.2, 2.3, 2.4, 2.5), выполняющая роль библиотеки типов объектов [3];

группа таблиц «3» (табл. 3.1, 3.2, 3.3, 3.4, 3.5), выполняющая роль библиотеки операций [3].

Строки таблиц, входящих в каждую из указанных групп, имеют общую (сквозную) порядковую нумерацию.

Следуя установившейся традиции, имена (коды) программ, типов объектов, экземпляров объектов и операций состоят из латинских букв и, как правило, представляют собой мнемонические обозначения, основу которых составляют некоторые слова или словосочетания на английском языке, которые тут же приводятся.

Таблицы группы «3», кроме всего прочего, содержат информацию о том, какие операнды использует та или иная операция и в какой роли (вход, выход, вспомогательный).

#### 4. Предметная деятельность (1-е измерение)

Выше, при обсуждении связанных процессов, в примере мы отправлялись от некоторого процесса А как от точки отсчета. Будем называть такую «отправную» деятельность «предметной». Применяя к деятельности прилагательное «предметная», мы тем самым признаемся в том, что будем строить нашу систему деятельности вокруг некоторого «эпицентра» целой предметной области. При этом мы, конечно, понимаем, что любая деятельность является так или иначе предметной, но почему бы нам не воспользоваться возможностью ввести условность, несколько упрощающую решение постоянно преследующих нас терминологических проблем?

Говоря о предметной деятельности, мы имеем в виду более или менее солидные пласты деятельности, относительно которых заранее можно быть уверенным, что к ним применяется программирование. И при этом нельзя или нецелесообразно обойтись обычными компьютерными программами – в процесс осуществления деятельности так или иначе «вплетается» человек и по тем или иным причинам требуется создавать и использовать программы деятельности, процесс выполнения которых сопровождается созданием его информационного образа (протокола).

Решая стоящую перед нами задачу поиска содержательного примера с учетом сказанного, мы обращаем внимание на предметную область создания полезных (качественных) объектов, то есть, объектов, удовлетворяющих конкретным качественным требованиям. Этой предметной области можно поставить в соответствие операцию предельно высокого уровня абстракции, на вход которой поступает информационный объект – описание требуемого качества, а выходом является объект, который предположительно этим качеством обладает. Соответственно мы можем начать заполнение наших таблиц типов объектов и операций (табл. 2.1 и 3.1).

Таблица 2.1. Типы объектов (начало)

№	Назначение типа	Код типа
1	Описание качества объекта (совокупность требований к качеству)	Qd (от англ. «quality descriptor»)
2	Объект, обладающий определенным качеством	Uo (от англ. «useful object»)

Таблица 3.1. Операции (начало)

№	Назначение операции	Код операции	Операнды					
			Входные		Выходные		Вспом-е	
			Имя	Тип	Имя	Тип	Имя	Тип
1	Создание полезного объекта	Суо (от англ. «create useful object»)	qd	Qd	Uo	Uo	od	Od

Таблица 2.2. Типы объектов (продолжение)

№	Назначение типа	Код типа
3	Описание объекта	Od (от англ. «object descriptor»)

Теперь перед нами встает задача определения способа выполнения введенной в рассмотрение операции. Это сделать несложно, поскольку при решении такого рода проблем всегда

возникает и используется промежуточная информация – описание искомого объекта, и исходная проблема решается в «два хода»: сначала по имеющимся требованиям к качеству объекта составляется его описание (например, конструкторская документация, включая чертеж), а затем по имеющемуся описанию (чертежу) изготавливается нужный объект. Пополним наши таблицы типов объектов и операций новой информацией (табл. 2.2 и 3.2).

Таблица 3.2. Операции (продолжение)

№	Назначение операции	Код операции	Операнды					
			Входные		Выходные		Вспом-е	
			Имя	Тип	Имя	Тип	Имя	Тип
2	Разработка описания объекта	Odd (от англ. «object descriptor design»)	qd	Qd	od	Od	-	-
3	Изготовление объекта	Om (от англ. «object manufacturing»)	od	Od	uo	Uo	-	-

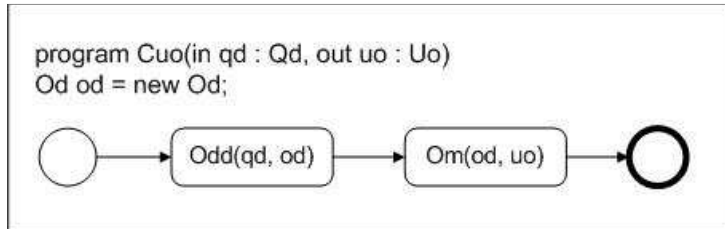


Рис. 1. Мета модель программы выполнения предметной операции

На рис. 1 представлена метамодель программы выполнения предметной операции.

В данном конкретном случае мы имеем дело с «фундаментальной», несмотря на ее примитивизм, метамоделью одного из самых важных человеческих видов деятельности. Конечно,

мы можем развить эту метамодель. Например, можем учесть, что для того, чтобы изготовить изделие в соответствии с конструкторской документацией, зачастую необходимо, кроме стандартного инструментария, иметь еще и специфические инструменты и приспособления (технологическую оснастку). Следовательно, между операциями разработки описания объекта и изготовления объекта можно вставить операции разработки и изготовления технологической оснастки. Кроме того, во многих случаях речь идет об изготовлении не одиночного изделия, а целых партий и т.д. Таким образом, для данной предметной области мы можем строить и развивать какие угодно программные метамодели – все зависит от того, что нам нужно.

Однако совершенно очевидно, что в силу возникшей на определенном этапе необходимости в каком-нибудь крупном предприятии данная метамодель программы, при должной проработке и детализации, плавно обретет статус программы деятельности, регламентирующей, скажем, деятельность конструкторского бюро и цеха опытных изделий. Это будет своя, специфическая для данного предприятия, программа деятельности.

То же самое может произойти и на другом, третьем и т.д. предприятиях. Каждое предприятие будет располагать своей собственной программой, но идея этих программ является одной и той же, будучи выраженной на уровне метамодели.

Таким образом, будем считать, что нам удалось проиллюстрировать соотношение метамодель-программа для аспекта деятельности, названного нами «предметным», и показать не только возможность, но и целесообразность их построения и использования на практике.

Примечание: рассмотрение предметного аспекта будет продолжено в разд. 6.

## 5. Организационная деятельность (2-е измерение)

Для того, чтобы предметная деятельность могла быть осуществлена, необходимо обеспечивать выполнение вводимых в план ее осуществления вспомогательных операций ресурсами и инициировать их выполнение. Типизация проблемных операций в зависимости от используемых ими ресурсов дает возможность рассмотреть особенности организации выполнения операций каждого типа и, в конечном итоге, разработать и использовать довольно несложную универсальную программу, регламентирующую организационную деятельность. Эта программа начинает работать, когда в БД описаний процессов появился очередной процесс. Добавим в табл. 2 новый тип объекта (табл. 2.3).

Таблица 2.3. Типы объектов (продолжение)

№	Назначение типа	Код типа
4	Описание процесса	Pd (от англ. «process descriptor»)

Определим организационную деятельность как операцию над описанием процесса и дадим предельно условную ее декомпозицию, основываясь на следующих допущениях:

1. Операции, выполнение которых необходимо организовать, различаются по типу:

- компьютерные (тип 'К');
- опорно-детализируемые с имеющейся программой (тип 'ОДП');
- опорно-детализируемые (тип 'ОД');
- остальные, требующие специфические ресурсы для своего выполнения.

2. Выполнение компьютерной операции может быть инициализировано сразу.

3. Имеющуюся программу опорно-детализируемой операции может немедленно начать использовать «исполнительный механизм» [3].

4. Если у опорно-детализируемой операции нет программы выполнения, придется ее создать применительно к данному конкретному случаю, используя имеющиеся или сконструировав новые вспомогательные опорные операции. После чего с приобретением программы она как бы меняет тип.

5. Остальные операции запускаются на выполнение только после предоставления им необходимых ресурсов.

В табл. 3.3 мы зафиксировали сказанное, добавив в предметную область новые операции.

Таблица 3.3. Операции (продолжение)

№	Назначение операции	Код операции	Операнды					
			Входные		Выходные		Вспом-е	
			Имя	Тип	Имя	Тип	Имя	Тип
4	Организация выполнения операции	Org (от англ. «organization»)	pd	Pd	pd	Pd	to	string
5	Определение типа операции	T_O (от англ. «type of operation»)	pd	Pd	to	string	-	-
6	Старт компьютерного процесса	Start_CP (от англ. «start of the computer process»)	pd	Pd	pd	Pd	-	-
7	Использовать программу	U_P (от англ. «use the program»)	pd	Pd	pd	Pd		
8	Создать программу	C_P (от англ. «create the program»)						
9	Назначение ресурсов	R_A (от англ. «resources assignment»)	pd	Pd	pd	Pd	-	-
10	Старт «реального» процесса	Start_RP (от англ. «start of the real process»)	Pd	Pd	pd	Pd	-	-

На рис. 2 представлена метамодель программы организации выполнения операций.

## 6. Нормализационная деятельность (3-е измерение)

Какой бы продуманной и организованной ни была человеческая деятельность, в процессах ее осуществления возможны различные отклонения от задуманного (запланированного),





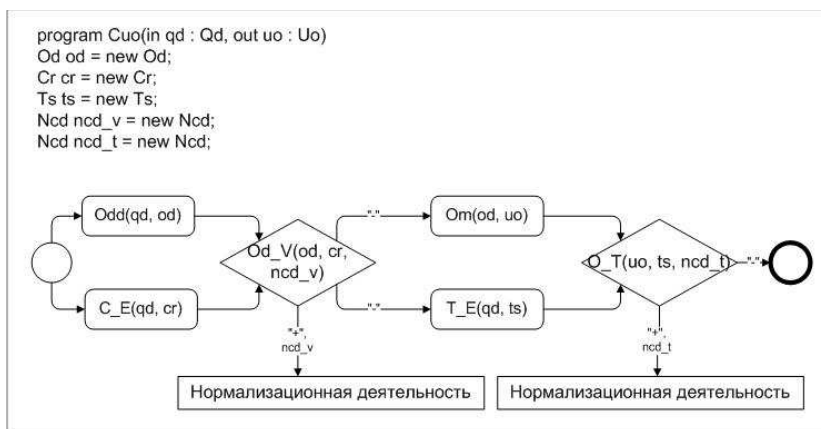


Рис. 3. Модифицированная метамодель программы выполнения проблемной операции

На рис. 3 представлена модифицированная программа выполнения исходной операции.

Если контроль результата выполнения предыдущей предметной операции дает отрицательный результат<sup>8</sup> ('-'), нормальное течение предметного процесса продолжается. Если же контроль дал положительный результат ('+'), то дальнейшие действия могут быть запрограммированы

лишь частично<sup>9</sup>, поскольку никогда нельзя сразу и с полной гарантией указать на истинную причину (или даже причины!) возникшего несоответствия. Более того, можно подвергнуть сомнению сам результат контроля<sup>10</sup>, поскольку в некоторых случаях из-за каких-то причин он может «поменять знак». Таким образом, можно сказать, что кроме всего прочего, положительный результат контроля прерывает нормальное течение предметного процесса и уводит нас в область, названную «нормализационной деятельностью». Следует обратить внимание, что на рисунке возврата в предметную деятельность нет. Но это не означает, что предметная деятельность прервана раз и навсегда (хотя в некоторых случаях может произойти и такое).

В соответствии с предложенной метамоделью могут быть созданы и использоваться различные программы, регламентирующие создание и контроль объектов, обладающих заданным качеством. Переход от метамодели к программе осуществляется за счет конкретизации:

<sup>8</sup> Как квалифицировать результат контроля (положительный/отрицательный) – зависит от того, что считать целью контроля. Если целью контроля считать доказательство соответствия продукта исходным требованиям, то результат положительный (отрицательный), если отклонения не обнаружены (обнаружены). Наоборот, если целью контроля считать нахождение возможных несоответствий, то результат положительный (отрицательный), если отклонения обнаружены (не обнаружены). Мы принадлежим к той части специалистов, которые твердо убеждены в справедливости и конструктивности второго подхода.

<sup>9</sup> Например, управление несоответствиями, требуемое [6] может быть основано на типовой обработке типовых несоответствий. Но если несоответствие не типовое, к нему требуется применить индивидуальный подход.

<sup>10</sup> Принято считать, что положительный и отрицательный результаты контроля носят разный характер в терминах «абсолютный/относительный». Для пояснения, о чем идет речь, и считая проверку всевозможных гипотез, в частности, поиск ошибок в доказательствах истинности математических утверждений одной из разновидностей контроля, на наш взгляд, уместно привести следующее высказывание Д. Пойа: «Математик, подобно натуралисту, проверяя некоторое следствие предполагаемого общего закона с помощью нового наблюдения, обращается с вопросом к природе: «Я подозреваю, что этот закон верен. Верен ли он?». Если следствие ясно опровергается, то закон не может быть верен. Если следствие ясно подтверждается, то имеется некоторое указание, что закон может быть верен. Природа может ответить «Да» или «Нет», но она шепчет один ответ и громко произносит другой: ее «Да» условно, ее «Нет» определено» [7]. Т.е., следуя предложенной логике, положительный результат («Нет») является абсолютным, отрицательный («Да») – относительным. Мы же стоим на позиции, с которой не можем абсолютизировать и положительный результат контроля.

вида создаваемых объектов;  
 правил описания требуемого качества объектов;  
 правил описания самих объектов;  
 используемых методов и средств изготовления объектов по имеющимся их описаниям;  
 используемых методов и средств контроля.

Совершенно очевидно, что программы, создаваемые в соответствии с предложенной метамоделью, обретут дополнительные уровни декомпозиции, на которых будут встречаться как автоматизируемые операции, так и неавтоматизируемые. Возможности автоматизации здесь зависят от того, насколько возможно автоматизировать процессы конструирования целевых объектов и конструирования вспомогательных объектов, необходимых для проведения их контроля.

Нормализационная деятельность начинается с операции, которую в первом приближении можно назвать «локализацией причин положительного результата контроля». Однако, поскольку эти причины могли или могут еще вызвать в том же процессе другие отклонения, правильнее было бы дать этой операции обобщенное название «локализация аномальной зоны процесса». Оче-

Таблица 2.5. Типы объектов (окончание)

№	Назначение типа	Код типа
8	Описание аномальной зоны процесса	Zd (от англ. «zone description»)

видно, что для расследований, направленных на локализацию аномальной зоны, большое значение имеет наличие протокола деятельности. Результат операции локализации заранее непредсказуем, поэтому можно считать, что получение положительного результата контроля – есть точка прерывания нормального течения процесса. Но, в общем случае, - не точка возврата в процесс. После локализации аномальной зоны процесса должны быть разработаны меры по нормализации процесса (ликвидации аномальной зоны). А это можно сделать только через перепланирование процесса. Возможно, при этом придется сделать некоторый «откат» назад в используемой программе, а, может, придется пересмотреть эту программу.

Итак, на основании сказанного мы можем в очередной (и последний) раз пополнить нашу коллекцию типов объектов и операций (табл. 2.5 и 3.5).

Таблица 3.5. Операции (окончание)

№	Назначение операции	Код операции	Операнды					
			Входные		Выходные		Вспом-е	
			Имя	Тип	Имя	Тип	Имя	Тип
15	Нормализация процесса	Norm (от англ. «normalization»)	ncd	Ncd	pd	Pd	zd	Zd
			pd	Pd				
16	Локализация аномальной зоны процесса	Loc (от англ. «localization»)	ncd	Ncd	zd	Zd	-	-
			pd	Pd			-	-
17	Перепланирование процесса	R_P (от англ. «replanning of the process»)	zd	Zd	pd	Pd	-	-
			pd	Pd			-	-

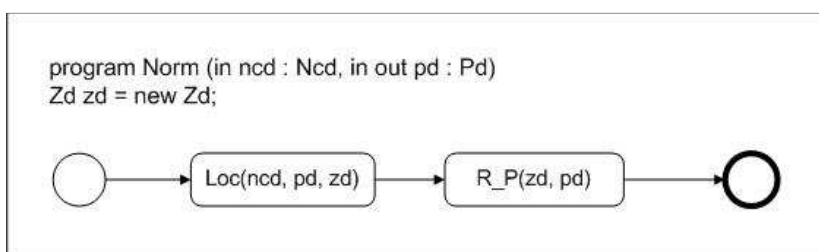


Рис. 4. Метамодель программы выполнения нормализационной операции

На рис. 4 представлена метамодель программы выполнения нормализационной операции.

Переход от данной метамодели к практически используемой программе, скорее всего, не приведет к ее видоизменению. Требуе-

мая конкретизация произойдет за счет фиксации форматов используемых информационных объектов. Сомнительной представляется также возможность дальнейшей декомпозиции используемых в метамодели операций. Описание способов их выполнения, являющихся прерогативой человека, вряд ли может быть выражено в терминах вспомогательных операций. Но если это окажется возможным в каком-то конкретном случае, остается только порадоваться процессу постепенного приближения к пониманию границ той деятельности, где человек ничем другим заменен быть не может.

## 7. Выводы

В заключение данной статьи, которая завершает небольшой цикл, посвященный обсуждению резервов программирования деятельности [2, 3], представляется целесообразным подвести итоги: как локальные (для данной статьи), так и для цикла в целом.

1. Феномен программирования имеет фундаментальную природу и проявляется лишь у человека, способного порождать и использовать информационные объекты, описывающие (фиксирующие) способы осуществления тех или иных определенных видов деятельности.

2. У человека нет причин ограничивать виды объектов, участвующих в программируемой им деятельности (входных, выходных, вспомогательных, преобразователей входов в выходы и т.д.).

3. Для многих видов человеческой деятельности важным является получение описаний процессов их осуществления (протоколов) с определенной степенью их детализации – до границы так называемого «опорного» уровня.

4. Программирование деятельности без ограничений на виды объектов и осуществление ее протоколирования требуют соответствующих средств создания и выполнения программ.

5. При наличии соответствующих средств последовательно осуществляемое программирование деятельности в конкретной предметной области должно приводить к ситуации, когда человеку отводится выполнение лишь тех действий, которые не способен выполнить компьютер (принятие нестандартных решений, творчество и т.д.).

6. Так называемое «моделирование бизнес-процессов» обычно проводится либо с целью изучения какой-либо деятельности для ее последующей автоматизации, либо для ее регламентации. В первом случае «модель» есть вспомогательный, промежуточный продукт, как правило, не имеющий самостоятельной ценности. Во втором случае «модель» есть конечный продукт, по существу, программа, предназначенная для выполнения людьми. Программирование же деятельности в описанном нами смысле в обоих случаях приводит к конечному продукту, который может выполняться с помощью компьютера, и относится к компетенции профессиональных программистов.

7. Программирование деятельности, примененное с целью метамоделирования программных аспектов связанных (через процессы осуществления) видов деятельности, позволило построить систему взаимосвязанных программных метамodelей, регламентирующих предметный и связанные с ним организационный и нормализационный аспекты деятельности. Эта система, которую, с учетом сделанных ранее оговорок, можно назвать 3-мерной программной метамodelью системы деятельности, может по указанным направлениям быть трансформирована в практически используемые программы деятельности в различных предметных областях.

## СПИСОК ЛИТЕРАТУРЫ

1. Фуксман А.Л. Технологические аспекты создания программных систем / Фуксман А.Л. – М.: Статистика, 1979. – 184 с.

2. Малышев О.В. Резервы программирования деятельности. Терминология / О.В. Малышев // Математичні машини і системи. – 2010. – № 1. – С. 150 – 161.
3. Малышев О.В. Резервы программирования деятельности. Система поддержки создания и использования программ деятельности / О.В. Малышев // Математичні машини і системи. – 2010. – № 3. – С. 84 – 95.
4. Котов В.Е. Введение в теорию схем программ / Котов В.Е. – Новосибирск: Изд-во «Наука», Сибирское отделение, 1978. – 258 с.
5. Business Process Modeling Notation (BPMN). Version 1.2 OMG Document Number: formal/2009-01-03 Standard document URL [Электронный ресурс]. – Режим доступа: <http://www.omg.org/spec/BPMN/1.2>.
6. ISO 9001:2008. Quality management systems. – Requirements. – Fourth edition 2008-11-15. – 36 p.
7. Пойа Д. Математика и правдоподобные рассуждения / Пойа Д. – М.: Наука, 1975. – 464 с.

*Стаття надійшла до редакції 19.11.2009*