ФОРМАЛИЗОВАННОЕ ОПИСАНИЕ ОБЩИХ СВОЙСТВ МОДЕЛИ[1]

С.В. Микони (Санкт-Петербург)

Введение

В связи с многообразием моделей, применяемых в имитационном моделировании и проблемой оценивания их качества [1], предпринимаются попытки формализации общих свойств модели. В работе [2] предложено формализованное определение модели кортежем символов < *O, N, Z, Is, L* > со следующим их содержанием: *O* – объект-оригинал, *N* – субъект моделирования (наблюдатель по Эшби), *Z* – цель моделирования, *Is* – инфраструктура моделирования, *L* – язык описания отношения объект-модель.

Предложенная авторами пятёрка символов описывает фактически не саму модель объекта *O* с присущими ей базовыми свойствами, а *систему моделирования*, включающую, в том числе, наблюдателя *N* и сформулированную им цель моделирования *Z*, с той поправкой, что *N* играет роль не наблюдателя, а *активного* участника процесса. Поэтому актуальной задачей является предложить формализованное определение собственно модели объекта, ещё не включённой в систему моделирования или уже отчуждённой от неё.

Модель языка предикатов первого порядка

Произвольное множество предикатов Р входит в состав многосортной алгебраической системы (структуры), представляемой четвёркой множеств [3]:

$$<A, C, F, P>, \qquad (1)$$

где: A – множество предметных переменных (носитель), C – множество констант и F – множество функций. Согласно [3] многосортная алгебраическая система является моделью языка предикатов первого порядка Ω. Её элементы – суть результаты интерпретации этого языка с применением функций *D, Cnst, Fn, Pr*:

$D: \pi \to A_\pi$, $\pi \epsilon Srt$;

$Cnst: cnst \to c$;

$Fn: fn \to f$;

$Pr: pr \to p$.

Элемент π множества *Srt* называется сортом объекта. Ему соответствует тип переменной в программировании. Для каждого сорта фиксируется множество предметных переменных $a^\pi_1,\ldots,a^\pi_n \epsilon A_\pi$ (носитель[2]) и констант $c^\pi_1,\ldots,c^\pi_k \epsilon C_\pi$. Множество констант C_π характеризует *диапазон значений* соответствующей переменной (домен в базе данных).

Функциональному символу *fn* сопоставляется n-местная функция (алгебраическая операция) $f: C_1 \times \ldots \times C_n \to C$. Задать п-арную ($n$-местную) операцию на множестве C – значит задать правило, которое любому упорядоченному набору из п элементов множества C ставит в соответствие однозначно определенный элемент того же множества C. Предикатному символу pr сопоставляется n+1-местный предикат p: $C_1 \times \ldots \times C_{n+1} \to$ B. В модели (1.1) он играет роль логической функции. Двоичный предикат имеет всего два значения: истина и ложь (true, false). В двоичной (булевой) логике они часто обозначаются через 1 и 0: $B=\{1, 0\}$. В нечёткой логике, предложенной Л. Заде, значения предиката могут принимать любые промежуточные значения между нулём и единицей: $B \epsilon [0, 1]$. Они трактуются как *частичная истина*. Поскольку предикат имеет смысл некоторого свойства Р объекта х, он может интерпретироваться функцией принадлежности $\mu_P(x)$, $0 \le \mu_P(x) \le 1$.

[1] Работа выполнена при финансовой поддержке РФФИ (проект № 17-01-00139).

[2] Носитель можно считать вырожденной алгебраической системой с пустым набором операций и отношений.

Содержательная разница между вычислительной и логической функцией (предикатом) обуславливается их ролью: первая применяется для *вычисления* значения переменной, а предикат – для *исчисления* истинности полученного значения. При этом возможен переход от n-местной функции $f(x_1,x_2,...,x_n)=y$ к n+1-местному предикату тогда и только тогда, когда предикат $p(x_1,x_2,...,x_n, y)$ истинен.

Функция принадлежности отражает частичную истину при классифицировании объектов. Например, принадлежность оцениваемой модели M на 80% классу моделей высокого качества представляется функцией принадлежности $\mu_{ДМ}(M)$=0,8.

В силу тождественности (изоморфизма) n-местного отношения R_c $C_1\times...\times C_n$ и n+1-местного предиката p: $C_1\times...\times C_{n+1} \to$ B, справедлива другая форма записи алгебраической системы <A, C, F, R>, в которой символ предикатов заменён на символ отношений R. Пару < F, R > называют *сигнатурой* системы. Использование только одного из этих символов представляют собой частные случаи алгебраической системы. К ним относятся соответственно реляционная система (модель) **B**= <A, C, R > и алгебра **A**= < A, C, F > [3].

Свойства модели

При интерпретации символов A и R реляционной системы множествами вершин V и связей E между ними (рёбер и/или дуг) переходим к модели графа $G = (V, C, E)$. Граф называется *помеченным*, если множество C содержит *имена* его вершин и связей, и *взвешенным* (нагруженным), если множество C содержит численные оценки вершин и связей. Графическое изображение реляционной системы является наглядной моделью любой структуры. Согласно этой роли помеченный граф представляет *структурную составляющую* модели, которую назовём **С-*моделью***.

Функции из множества F в алгебре **A** = < A, C, F > принято называть операциями. Свойства операций определяются относительно элементов носителя A, являющихся их аргументами. Элементами носителя могут быть числа, которые делятся на *натуральные* **N**, *целые* **Z**, *рациональные* **Q**, *вещественные* (действительные) **R**. Относительно числа операций, соблюдения сочетательного (ассоциативного) и перестановочного (коммутативного) законов, наличия нейтрального и обратных элементов алгебры имеют различную вычислительную мощность.

Любая функция представляет собой отображение области её определения в область значений. Представим носитель A множествами определения и значений функции f: $A=XUY$. Пусть функция f отображает в область значений Y n аргументов $x_1,...,x_n$: $y=f(x_1,...,x_n)$. Если нас не интересуют внутренние связи между аргументами отображения, выражение $y=f(x_1,...,x_n)$ представляет собой функциональную модель (**Ф-модель**).

В общем случае необходимо отображать как структуру, так и функции объекта-оригинала, что влечёт объединение С- и Ф-моделей в структурно- функциональную модель (**СФ-модель**). Однако представление объекта-оригинала СФ-моделью ещё не означает решения задачи моделирования. Результат моделирования можно получить только в том случае, если модель ***разрешима***. Этой проблеме, в частности, посвящена работа [4]. Применительно к численному решению задачи *функция называется* ***вычислимой***, *если существует вычисляющий её алгоритм*. Обобщая вычислимость на параллельные вычисления, назовём модель, раскрывающую функцию через совокупность последовательных и/или параллельных *операций* ***О-моделью***.

Исходя из теории разрешимости, вычислимой является модель, объединяющая Ф- и О-модели, а в общем случае – СФ- и О-модели (СФ-модель & О-модель). С точки зрения разрешимости С-модель не реализуема в отсутствие Ф-модели, а Ф-модель, в свою очередь, не реализуема в отсутствие О-модели. На принципе объединения этих моделей основаны динамические библиотеки в программных системах. Иными словами, минимальной полнотой обладает Ф-модель в сочетании с О-моделью, как модель реализуемого чёрного ящика (минимальный состав *D*-модели).

СФ-модель содержит информацию о том, ***какие*** свойства объекта-оригинала моделируются, а О-модель – ***как*** решить задачу моделирования. Однако описать свойства и способ решения задачи не означает способность решить задачу моделирования. По этой причине СФ- и О-модели, не подготовленные к исполнению в среде моделирования, называются *описательными* или *D*-моделями (от *Descriptive*). Противоположностью описательной модели является *исполнимая* модель или *E*-модель (от *Executable model*), подготовленная для реализации в среде моделирования.

Примером *D*-модели в программировании является текст программы, а примером ***E***-модели – компилированная программа, готовая к выполнению. Соответствующие файлы имеют расширения языка программирования и выполнения (*.exe от executive). Выполняемая версия модели активна в породившей её системе моделирования.

Таким образом, исходя из формального описания модели в языке предикатов первого порядка четвёркой $<A, C, R, F>$, можно утверждать, что для выполнения моделирования модель должна содержать *структурную, функциональную* и *операционную* составляющие. В совокупности они и представляют собой перечислительное определение модели, отражая свойства, присущие любой модели. Из них функциональная (Ф-модель) и операционная (О-модель) составляющие обязательны для выполнения моделирования. Структурная составляющая (С-модель) может опускаться при целостном представлении объекта моделирования. Таковы, например, модули динамической библиотеки, воспринимаемые как целостные объекты.

В силу своей общности формальная модель может рассматриваться как *прообраз* модели любого назначения. Иными словами, любая модель должна *наследовать* свойства структуры, функций и способа их выполнения, трактуя их в соответствии со своим назначением. Несмотря на то, что выполнимость модели требует применения более, чем одной составляющей, представляет интерес изучение свойств каждой составляющей в отдельности в сопровождении соответствующих примеров [5]. Реальные модели именуются именем той составляющей, значение которой первично для моделирования.

Структурная модель

С-модель востребована для анализа структурного подобия между моделью и объектом-оригиналом. Она наследует реляционную систему **B**, отражая связи между элементами системы. Элементы с именами из множества *C* принадлежат носителю *A*, а связи между элементами принадлежат отношению смежности R_c:

$$M_c=< A, C, R_c >. \qquad (2)$$

С-модель представляется помеченным графом. Метками являются имена вершин и дуг (рёбер) графа. Граф, в котором метками служат переменные, инвариантен относительно предметных областей.

Ориентированный граф *G* задаёт *последовательность обработки* информации (энергии, вещества), и, следовательно, отображает фактор времени. В отличие от него в неориентированном графе (неорграфе) не представляется возможным отделить входы от выходов, а, следовательно, и определить *последовательность* обработки информации (энергии, вещества).

Примерами структурной модели, представленной орграфом, являются дерево целей и алгоритм вычислений. Неориентированным графом описывается, например, компьютерная сеть. Объект, характеризуемый несколькими свойствами (атрибутами), представляется звёздным графом. Между собой сущности связываются рёбрами. Структурные модели такого вида применяются в составе моделей реляционных баз данных, называемыми диаграммами «сущность-связь» (ER-диаграммы). Учитывая разные свойства вершин в диаграмме «сущность-связь», эта модель является неоднородной.

В программировании С-модель именуется *структурой данных*. *ER*-диаграмма, например, представляет умозрительную форму структуры данных. Для её машинного представления применяется унифицированный объектно-ориентированный язык моделирования UML (Unified Modeling Language).

В терминологии искусственного интеллекта структура данных называется моделью-*прототипом,* что объясняется отсутствием в ней значений переменных. Каждая переменная может получить значение из своей области определения (домена).

Функциональная модель

Функциональная модель (Ф-модель) наследует алгебру $\mathbf{A} = < A, F >$, если носитель A представить множествами X и Y, где множество X означает область определения, а множество Y – область значений функции $f \epsilon F$. Функция f представляет собой отображение $f: X \rightarrow Y$. Исходя из принятых обозначений, Ф-модель описывается тройкой символов:

$$M_{ф} = < X, Y, F >, \tag{3}$$

где X U $Y = A$. В частном случае $X = Y = A$.

Выразим отображение $f: X \rightarrow Y$ через функциональное отношение между переменными x и y: $y = f(x)$. Если переменные x и y имеют конечное число значений k, такое, что $k < |X|$ и $k < |Y|$, то для представления области определения X функции f следует применить $n = \log_k |X|$ k-значных входных переменных, а для представления области значений функции f потребуется $m = \log_k |Y|$ k-значных выходных переменных, причём $n \neq m$, если $|X| \neq |Y|$, и $n=m$, если $|X| = |Y|$.

Применительно к машинным кодам, выраженным через двоичные переменные ($k=2$), области определения и значений функции f, заданные одним байтом, имеют 256 значений. Для их представления требуется $n = m = \log_k 256 = 8$ переменных, что и соответствует восьмиразрядной двоичной сетке (байту). Входная или выходная информация, содержащаяся в байте, представляются соответствующими векторами входных или выходных переменных: x=($x_1, \ldots, x_8$) или y=($y_1, \ldots, y_8$). В векторной форме функциональная зависимость описывается формулой $y=f(x)$.

Функция f может иметь любую сложность и выражаться через функционал. Ф-модель определяет, лишь, способ задания соответствия между областями определения и значений функции. По этой причине она именуется *«чёрным ящиком»* – известно «что» получается, но неизвестно «как». Пользователь обычно и воспринимает объект использования как «чёрный ящик». Ему безразлично, *как* устроен объект. Важно, что можно получить на его выходах, задав входное воздействие.

В случае $|F|= 1$ модель «чёрного ящика» называется однофункциональной, а при $|F|>1$ – многофункциональной. Примером многофункциональной модели является система команд компьютера.

Ф-модель системы S, находящейся под воздействием со стороны предшествующей системы, субъекта управления и внешней среды, описывается пятёркой:

$$M_s = < X_{вх}, X_y, X_{вс}, Y, F >. \tag{4}$$

В модели (4) область определения $X = X_{вх} \cup X_у \cup X_{вс}$ представлена тремя составляющими, соответствующими трём видам воздействия на систему: входным, управляющим и воздействиям внешней среды.

В отношении вида обрабатываемой информации Ф-модели делятся на модели: *детерминированные и недетерминированные, непрерывные* и *дискретные*. К классу недетерминированных Ф-моделей относятся *стохастические, нечёткие* и *интервальные* модели, отражающие различные виды неопределённости.

Операционная модель

Ф-модель, задавая правило преобразования входных воздействий в выходные реакции, не объясняет, *как* оно выполняется. Между тем, любые простейшие операции, даже такие как сложение, не выполняются одномоментно, представляя собой последовательность элементарных действий. Эта последовательность, обладающая свойствами определённости, конечности числа шагов, массовости и результативности, названа *алгоритмом.* Заметим, что понятие алгоритма необязательно относится к компьютерным программам, так, например, технологическая карта представляет собой алгоритм технологического процесса.

Пошаговое выполнение алгоритма влечёт изменение внутреннего состояния объекта действия – от $q(t)$ к $q(t + 1)$. Эти изменения привязываются к моментам времени t=0, 1, 2, …, где t=0 фиксирует начальное состояние объекта. Закономерность смены состояний описывается *функцией переходов* $f_п$, дополняющей *функцию выхода* объекта $f_в$.

Полагая $f_п$, $f_в \epsilon F$ и $q \epsilon Q$, получаем следующее расширение Ф-модели, соответствующее модели конечного автомата:

$$M_{KA} = < X, Y, Q, f_п, f_в, t > \tag{5}$$

В векторной форме модель конечного автомата $M_{КА}$ записывается как:

$q(t)=f_п(q(t – 1), x(t)),$

$y(t)=f_в(q(t), x(t))$ – для автомата Мили и

$y(t)=f_в(q(t))$ – для автомата Мура.

В теоретико-множественной форме функции переходов и выходов КА имеют вид:

$f_п$: $X \times Q \rightarrow Q$;

$f_в$: $X \times Q \rightarrow Y$ (автомат Мили);

$f_в$: $Q \rightarrow Y$ (автомат Мура).

В соответствии с разделением времени t на такты t – 1, t, t + 1,… модель конечного автомата M_{KA} является *дискретной*. Графически она описывается графом переходов, отражающим смену внутренних состояний при внешних воздействиях и формирование реакций на них.

В отличие от Ф-модели модель конечного автомата M_{KA} характеризует не правило, а *процесс* реализации функции, т.е. является её О-моделью. Последовательность выполнения операций, выраженная через символ времени t, позволяет отнести О-модель к классу динамических моделей.

Структурно-функциональная модель

Согласно своему названию структурно-функциональная модель (СФ-модель) совмещает свойства структурной и функциональной моделей. Для преобразования С-модели в СФ-модель метки, входящие в носитель структуры, ассоциируются с функциями, реализуемыми Ф-моделями, входящими в СФ-модель.

Поскольку элементами структуры являются вершины (vertex) и ребра (edge) графа, для ассоциации меток с реализуемыми ими функциями используются отношения $R_{fv} = \{(a_i, f_i)\}$, $i = \overline{1,n}$, $R_{fe} = \{(a_i, a_j), f_{ij}\}$, $i, j = \overline{1,n}$, $i \neq j$. Функции СФ-модели рассматриваются с точки зрения их роли в модели.

С учётом объединения С-модели с Ф-моделью и разделения множества отношений на 3 вида $R = R_c \cup R_{fv} \cup R_{fe}$ СФ-модель имеет вид:

$$M_{сф} = < A, R_c, R_{fv}, R_{fe}, C, F >. \qquad (6)$$

Для построения СФ-модели на основе С-модели необходимо дополнительно сформировать множество функций F и два множества ассоциаций R_{fv} и R_{fe}.

Согласно терминологии искусственного интеллекта модель $M_{сф}$, не содержащая констант, является СФ-моделью-*прототипом*. Модель-прототип с означенными переменными представляет собой СФ-модель-*экземпляр*. Примером СФ-модели в объектно-ориентированном программировании является *класс*, объединяющий структуру данных и операции над ними.

Подчеркнём диалектическую взаимосвязь между Ф- и СФ- моделями. Ф-модель отражает целостные (внешние) свойства объекта, а СФ-модель отражает внутренние свойства целого при раскрытии «чёрного ящика». Различие между ними заключается во взгляде на объект моделирования – извне или изнутри.

СФ-модель отражает только причинно-следственное отношение на множестве входящих в неё Ф-моделей. В ней отсутствует фактор времени t. Следовательно, так же, как и Ф-модель, она нуждается в операционной составляющей. Однако её роль не может играть О-модель, даже при расширенной (параллельной) трактовке алгоритма. Это объясняется тем, что параллельная трактовка алгоритма не касается непредсказуемой состязательности процессов в реальных системах.

Модель, совмещающую описательный и операционный аспекты, представляемые соответственно СФ- и О- моделями, назовём системной моделью (S-моделью):

$$M_s = < A, R_c, R_{fv}, R_{fe}, C, F, t >. \qquad (7)$$

Исследования, выполненные по данной тематике, проводились в рамках бюджетных тем №№0073–2014–0009, 0073–2015–0007.

Выводы

1. Выделение общих свойств модели облегчает анализ качества моделей любого назначения, применяемых в системе имитационного моделирования.

2. Для анализа связей используется С-модель. Для оценивания функций используется Ф-модель. Для анализа выполняемых операций используется О-модель. Для комплексного оценивания свойств используется СФ-модель.

3. Деление моделей на описательные и исполнимые позволяет оценивать качество как ручных, так и машинных моделей.

Литература

1. **Соколов Б.В., Юсупов Р.М.** Концептуальные основы оценивания и анализа качества моделей и полимодальных комплексов // Известия РАН. Теория и системы управления, 2004, №6. С. 5–16.
2. **Волкова В.Н., Козлов В.Н., Магер В.Е., Черненькая Л.В.** Классификация методов и моделей в системном анализе. // Сборник докладов XX Международной конференции по мягким вычислениям и измерениям (SCM-2017). 24-26.05.2017. – СПб.: СПбГЭТУ (ЛЭТИ). 2017. С. 223–226.
3. **Курош А.Г.** Лекции по общей алгебре. СПб. Лань. 2007. 416 с.
4. **Хованский А.Г**. Топологическая теория Галуа. Разрешимость и неразрешимость уравнений в конечном виде. М.: Изд-во МЦНМО, 2008. 296 с.
5. **Микони С.В.** Общие диагностические базы знаний вычислительных систем. СПб.: СПИИРАН. 1992. 234 с.