

УДК 519.68

Современные проблемы математического моделирования в исследовании операций

А.Н. Рыжаков*, О.А. Щербина**

* Крымский институт бизнеса,
Симферополь 95000. *E-mail*: AR1961253@yandex.ru

** University of Vienna,
Vienna 1090, Austria. *E-mail*: oleg.shcherbina@univie.ac.at

Аннотация. Рассмотрены методологические проблемы математического моделирования в исследовании операций, в том числе приведена классификация моделей, обоснована необходимость этапа построения концептуальной модели при использовании комплекса моделей, а также необходимость интеграции моделей, описывающих разные подсистемы. Сделан обзор и описаны основные черты структурного моделирования, являющегося весьма перспективным подходом, позволяющим решить эти задачи.

1. Введение

Исследование операций является дисциплиной, позволяющей на основе применения математических моделей находить рациональные, научно обоснованные решения. Использование математических моделей непрофессиональными пользователями затруднено в связи с отсутствием у них должного математического образования. Кроме того, это объясняется отсутствием достаточно общих эффективных и в то же время интуитивно понятных подходов моделирования.

При использовании математического моделирования исследуемый исходный объект заменяется его математической моделью, которая затем исследуется с помощью современных методов вычислительной математики и применения компьютерной техники. Методология математического моделирования интенсивно развивается, охватывая все новые сферы от разработки больших технических систем и управления ими до анализа сложнейших экономических и социальных процессов. Математические модели применяются в системах поддержки принятия решений (СППР)[1], [3], позволяющих непрофессиональному пользователю компьютера использовать современные математические модели и методы исследования операций при поиске оптимального решения. Можно говорить об эволюционном процессе „смены” парадигм моделирования. При этом первый этап моделирования характеризовался формализацией достаточно простых моделей реальных объектов с линейными зависимостями и малой размерностью. Методы анализа моделей заключались в основном в получении соответствующих аналитических решений. Второй

этап характеризовался попытками возможно более полного описания объектов реального мира, причем объект представлен в модели в виде сложной системы, при этом учитывалась нелинейность зависимостей, размерность моделей возросла и достигла нескольких десятков. Для записи моделей в компьютере и решения соответствующих вычислительных задач используются языки программирования. Важную роль на этом этапе приобретает вычислительный эксперимент [2]. В настоящее время можно говорить о третьем поколении математических моделей, в которых используются методы автоматизации формирования моделей и данных, возникают языки моделирования [25], существенно отличающиеся от обычных языков программирования своей декларативной природой.

Моделирование реальных сложных систем на базе современных компьютерных технологий делает актуальным внедрение сред моделирования [9], позволяющих переложить многие рутинные задачи цикла моделирования на компьютер. В настоящее время чрезвычайно актуальны вопросы автоматизации различных аспектов математического моделирования: выбора и построения моделей, записи моделей в форме, близкой к обычной математической записи, создание библиотек решателей (решатель (solver) — программное обеспечение, предназначенное для решения рассматриваемой математической задачи).

Анализ последних достижений и публикаций, посвященных проблеме математического моделирования [2], позволяет сделать вывод, что в настоящее время в отечественной литературе практически отсутствуют публикации по автоматизации моделирования, что и обусловило необходимость подготовки настоящего обзора.

Целью настоящей работы является обзор ряда направлений автоматизации математического моделирования, в том числе методов структурного моделирования, практически неизвестного в отечественной литературе.

2. О процессе моделирования

Остановимся на терминологии моделирования. Различаются „моделирование в малом” и „моделирование в большом” [26]. Первое концентрируется на паре „проблема — модель”: концептуализация, формулировка и проверка адекватности конкретной модели для решения данной задачи. Моделирование в большом рассматривает организацию системы управления библиотекой совместно используемых моделей.

Основу математического моделирования составляет триада модель — алгоритм — программа [2]. Рассмотрим подробнее указанные составляющие. На первом этапе выбирается (или строится) модель исследуемого объекта, отражающая в математической форме важнейшие его свойства — законы, которым он подчиняется, связи между его составляющими элементами, и т. д. Под моделью при этом понимается «„эквивалент” объекта, отражающий в математической форме важнейшие его свойства — законы, которым подчиняется, связи, присущие составляющим его частям, и т.д.» ([2, с. 8]).

Само по себе построение моделей представляет собой «применение фундаментальных законов природы, вариационных принципов, аналогий, иерархических цепочек» ([2, с. 11]), а процесс построения моделей включает в себя следующие этапы ([2, с. 25–26]):

1. «Словесно-смысловое описание объекта или явления» («формулировка пред-модели»);
2. «Завершение идеализации объекта» и упрощение описания;
3. Переход «к выбору или формулировке закона (вариационного принципа, аналогии и т.п.)» и его записи в математической форме;
4. «Завершает формулировку модели ее „оснащение”» (задание начального состояния и параметров объекта). Этот этап особенно важен, поскольку: «...на-конец, формулируется цель исследования модели ...»;
5. Модель изучается всеми доступными методами (в том числе с применением различных подходов и вычислительных методов);
6. В результате исследования математической модели достигается поставлен-ная цель. При этом «должна быть установлена всеми возможными способами (сравнением с практикой, сопоставлением с другими подходами) ее адекват-ность — соответствие объекту и сформулированным предположениям...».

Заметим, что описанные выше этапы 1–2 моделирования часто называются в лите-ратуре построением *концептуальной модели*. Концептуальная модель определяет элементы, структуру и границы моделируемой системы, свойства ее существенных элементов, связи между элементами, существенные для моделируемой системы (с точки зрения цели системы или цели моделирования). Цель концептуального моде-лирования — сделать объекты реального мира доступными на абстрактном уровне, а также помочь исследователю в исключении не относящихся к делу элементов и связей. Концептуальные модели могут представлять комбинацию текстовых опи-саний, таблиц, параметров, диаграмм.

Концептуальные модели описывают существенные компоненты и процес-сы системы, документируют предположения о характере связей компонент и процессов между собой, а также выявляют пробелы в знаниях о моде-лируемой системе — т.е. они являются рабочими гипотезами о форме и функции моделируемой системы [22].

Как правило, разработка концептуальной модели производится перед разра-боткой более детализированной формальной математической модели.

Роль концептуального моделирования возрастает при создании комплекса мо-делей, который описывает сложную систему.

3. Уровни абстракции моделей

Попробуем описать возможные различия в степени абстракции моделей.

В [9] выделяется 4 уровня абстракции для моделей. На самом нижнем уровне находится „конкретная модель” (model instance), которая является *конкретизацией модели*, получающейся путем присвоения параметрам модели их числовых значений (или, согласно [2], путем „оснащения” модели данными). Следующий, более высокий уровень абстракции — *класс моделей*, который является множеством подобных моделей с известной математической формулировкой без конкретных числовых данных. Третьим уровнем абстракции является *парадигма моделирования*, которая является множеством классов подобных моделей. И, наконец, на четвертом уровне абстракции находится „традиция моделирования” (*подход моделирования*), которая является группой схожих парадигм моделирования и может быть разделом прикладной математики. Например, исследование операций — традиция моделирования, в которую входят такие парадигмы моделирования, как имитационное моделирование, теория массового обслуживания, математическое программирование, теория марковских процессов. Каждая парадигма исследования операций содержит свои классы моделей. Так, например, в рамках парадигмы математического программирования рассматриваются классы моделей линейного и нелинейного программирования, стохастического программирования и т.д.

4. Об интеграции моделей

В течение последних десятилетий интенсивно развивались различные парадигмы моделирования. При этом каждая парадигма моделирования включает систему накопленных знаний, опыта, методологии и средств моделирования, предназначенных для решения соответствующих этой парадигме задач.

Парадигма (от греч. paradeigma пример, образец) — общие принципы, концептуальная схема постановки проблем и их решения, методов исследования, общепринятых в научном сообществе [19], [23]. При этом научная парадигма состоит из теорий, законов, правил, моделей, концепций и определений, общепринятых в научном сообществе.

Парадигмы моделирования различаются по типу моделей (например, статические, динамические, непрерывные, дискретные, детерминированные, стохастические, нечеткие, модели с „мягкими ограничениями”), по различным методам анализа моделей исследования операций (например, имитация, оптимизация, многокритериальная оптимизация, нечеткая оптимизация). Кроме этого, в рамках тех или иных парадигм моделирования для решения соответствующих задач разрабатываются специализированные решатели.

При моделировании реальных и достаточно сложных систем вряд ли окажется возможным и достаточным создание лишь одной модели, адекватно описывающей все подсистемы моделируемой системы и их взаимодействие. Подобная всеобщая модель была бы слишком сложна как для понимания, так и для отладки. Обычно

сложная система может быть описана с помощью комплекса моделей, возможно принадлежащих различным парадигмам или даже традициям моделирования. Понятна актуальность интеграции моделей при решении этих задач. Мультипарадигмальное моделирование [23], определяемое как эффективное приложение всех подходящих парадигм моделирования, является одним из важных аспектов моделирования сложных систем. В некоторых ситуациях возможно использовать более общую парадигму, содержащую более простую парадигму (например, построение нелинейной модели, содержащей линейную часть). Однако даже в этом случае обычно более рационально использовать нелинейную и линейную парадигмы для соответствующих частей этой модели). Итак, вместо использования более общей парадигмы, более рационально использовать объединяющую парадигму, например, парадигму, которая работает с нелинейными моделями с линейной частью. В других ситуациях может оказаться необходимым *переключение парадигм*.

Интеграция различных моделей исследования операций при моделировании сложных реальных систем часто бывает затруднительной, поскольку эти модели построены в рамках различных парадигм моделирования. Одним из наиболее известных и перспективных подходов интеграции различных парадигм моделирования является структурное моделирование [10], [11], [23].

В литературе [15] выделяется глубокая и функциональная интеграция моделей. *Глубокая интеграция* строит одну новую модель, объединяющую 2 или более данных моделей при выполнении важного условия, что новая модель должна быть представлена в той же формальной системе определения моделей, что и исходные модели. Под формальной основой определения моделей здесь понимается либо традиция моделирования, либо парадигма моделирования.

Функциональная интеграция, напротив, не строит новую модель на той же формальной основе. Она оставляет эти модели в их первоначальной форме и накладывает некоторую вычислительную схему для координации вычислений по этим моделям, обычно направляя определенные выходы одних моделей к входам других моделей, задавая тем самым порядок вычислений. Схема функциональной интеграции может быть выражена на языке взаимосвязи моделей.

В работе [9] отмечена зависимость вида интеграции от того, какие именно, по степени абстракции модели должны быть интегрированы. В результате могут быть выделены десять возможных типов интеграции, которые предусматривают, например, объединение двух или более традиций, парадигм или классов моделирования.

5. Основы структурного моделирования

5.1. Языки и системы моделирования

Структурное моделирование было предложено А.Джефффрионом [10] в качестве унифицированной концептуальной основы моделирования, основанной на использовании бесконтурных *орграфов* (ориентированных графов) для представле-

ния взаимных связей между элементами модели и иерархических структур для представления уровней абстракции, для облегчения и интеграции различных аспектов моделирования.

Язык моделирования — формальная компьютерная исполнимая нотация (система обозначений), которая может быть использована для выражения абстрактных понятий концептуальной основы моделирования (специализации в подходе моделирования).

Для выражения абстрактных понятий структурного моделирования разработан язык SML [11], [12].

Языки моделирования хранят *знание* о модели, они *определяют задачу* и обычно не определяют, как решать ее. Это — *декларативные* языки, описывающие модель путем определения ее свойств. К трем наиболее важным свойствам декларативных языков можно отнести следующие: 1) Модели описываются декларативным образом; 2) четкое разделение между описанием модели и процессом решения; 3) четкое разделение между структурой задачи и данными.

Структурное моделирование является концептуальной основой — разновидностью графового подхода. Графовый подход к управлению моделями использует понятия теории графов для представления математической модели. Согласно этому подходу, математическая модель представляется с помощью одного или более графов или орграфов. Использование графов при представлении знаний обладает известными преимуществами, такими, как концептуальная ясность, легкость программирования и простота манипулирования. Графовая модель состоит из вершин и дуг, отражающих семантику модели. Представление сложных математических моделей посредством графов приводит к более эффективному диалогу между специалистами по исследованию операций и заказчиками. Представление задачи оптимизации в виде структурной модели является бесконтурным орграфом, разбитым на части и описывающим все элементы задачи и соотношения между ними.

5.2. Элементная структура

Структурное моделирование (СМ) рассматривает модель, состоящей из дискретных элементов. При этом предполагается, что каждый элемент либо задается в виде определения первоначального объекта, либо выражается через другие, ранее определенные элементы. Элементная структура стремится охватить все детали определения конкретного примера модели. Она может рассматриваться в виде орграфа, состоящего из вершин (элементов) и дуг („вызовов”). Каждый вызов представляет ссылку на элемент при определении некоторого другого элемента. Исходная вершина каждой дуги представляет собой вызывающий элемент, а конечная вершина — вызываемый элемент.

Можно выделить пять типов элементов:

1. Элементы - первоначальные объекты (primitive entity elements) не имеют присвоенных им значений и в общем представляют объекты или понятия,

- заданные как первоначальные для модели (например, белок как питательное вещество);
2. элементы - составные объекты (compound entity elements) не имеют присвоенных им значений и в общем представляют объекты или понятия, определяемые через другие, ранее определенные элементы (например, „связь” в транспортной задаче определяется заданием определенного поставщика и определенного потребителя);
 3. элементы-атрибуты (свойства, признаки) имеют постоянное значение и обычно представляют свойства объектов или понятий (например, минимальная суточная потребность в белке, в г);
 4. элементы-функции имеют значение, зависящее от определенного правила действия над значениями вызываемых элементов, и в общем представляют вычисляемые свойства и более сложные аспекты моделей (например, общие годовые расходы на хранение);
 5. элементы-тесты подобны элементам-функциям, разница в том, что первые принимают лишь логические значения («истина», «ложь»).

Если граф представляет всю элементную структуру, то он должен иметь атрибуты, которые приписаны вершинам графа и его дугам для того, чтобы представить значения необъектных элементов; правила, по которым вычисляются значения элементов-функций и элементов-тестов; порядок для входящих вершин в каждую вершину.

5.3. Родовая структура

Родовая структура (generic structure) описывает группировки элементов, что достигается путем разбиения множества всех элементов на роды по типам элементов, каждому типу соответствует подмножество разбиения. Тогда каждый род (genus) включает элементы одного и того же типа (например, множество всех первоначальных элементов, представляющих питательные вещества).

Допустимое разбиение должно удовлетворять свойству родового подобия, которое означает, что каждый элемент, принадлежащий роду, вызывает элементы *из одного и того же* другого рода. По существу, родовой граф является *конденсированным графом* [17] элементного графа.

5.4. Модульная структура

Модульная структура иерархически организует родовую структуру удобным образом. Основная идея состоит в группировке родов в концептуальные объекты, называемые модулями, согласно их семантической общности, затем в объединении этих модулей в модули более высокого порядка и т.д. (например, род «питательные

вещества» и род «минимальные дневные потребности» могут быть объединены в модуль «данные о питательных веществах»).

Модульная структура представляет собой корневое дерево, корень которого представляет всю модель, а листья однозначно соответствуют родам. Остальные вершины дерева являются модулями, представляющими концептуальные объекты, состоящие из их родов — потомков. Допустимая модульная структура должна разрешать представление в виде списка с отступами в порядке без прямых ссылок, т.е. все рода должны быть перечислены в таком порядке, чтобы ни один элемент из рода не вызывал элемент рода, который расположен далее и ниже в списке. Модульная структура, удовлетворяющая этому условию, называется монотонной, а ее список с отступами называется *модульной схемой*.

В практических задачах обычно нетрудно и естественно определить монотонную модульную структуру. Если по какой-либо причине это сделать не удастся, всегда есть возможность построить тривиальную монотонную модульную структуру, состоящей лишь из корня и всех родов, упорядоченных согласно топологической сортировке родового графа.

5.5. Структурная модель

Структурная модель — это (а) элементная структура плюс (б) родовая структура плюс (с) монотонная модульная структура.

Необходимо также отметить, что СМ не ограничено статическими моделями. Динамические модели с дискретным временем могут быть приспособлены к этой схеме путем введения рода первоначальных объектов с числом элементов, равным числу временных промежутков, а динамические модели с непрерывным временем часто могут быть адаптированы, если ввести функции атрибутов (параметров) и функции элементов, зависящие от времени.

5.6. Схема модели

При моделировании реальных объектов редко имеют дело с единичным примером модели, а скорее с классом подобных моделей. В связи с этим, вводится удобная формализация класса «подобных» структурных моделей — так называемая *схема модели*. Схему модели неформально можно описать как любой класс структурных моделей, модульные схемы которых находятся во взаимнооднозначном соответствии по отношению друг к другу таким образом, что это совместимо с модульной структурой, с родовой структурой и с назначением моделей.

6. Пример и дополнительные понятия

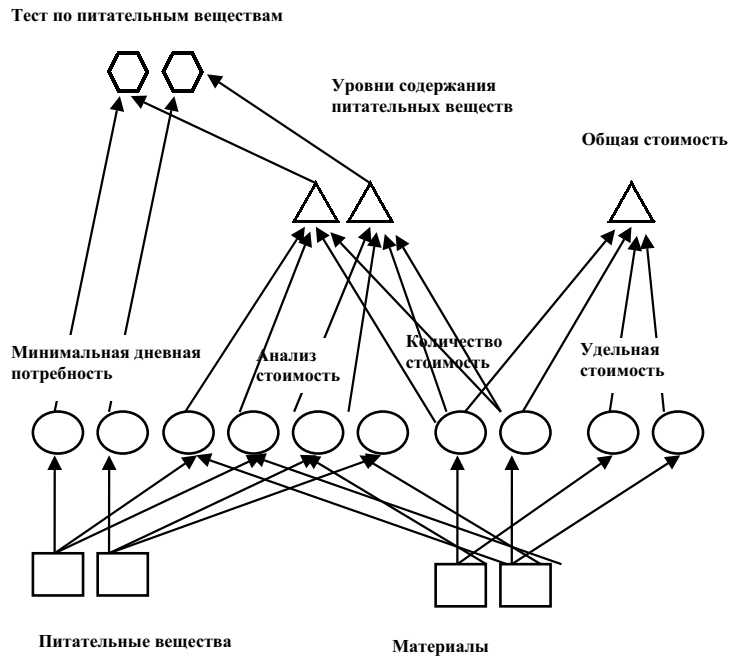


Рис. 1. Элементный граф для задачи о рационе питания.

6.1. Элементный граф

На рис.1 изображен *элементный граф* простой модели с двумя продуктами питания и двумя материалами, из которых составляются варианты (рационы) питания. Это оргграф элементной структуры без каких-либо пометок, характеризующих атрибуты вершин или дуг. Рис. 1 использует неформальные комментарии для указания типа вершины и ее содержательного смысла. Тип вершины задается здесь графическим символом: квадраты обозначают первоначальные объекты, круги соответствуют атрибутам (параметрам), треугольники — элементам-функциям, шестиугольники — элементам-тестам. Здесь нет составных элементов. Напомним, что дуга выражает вызов конечного элемента исходным элементом. Процесс вычисления значений всех элементов-функций и элементов-тестов в их естественном топологическом порядке называется оценкой.

6.2. Родовой граф

Естественным образом элементы разбиваются на группы. Элементы - первоначальные объекты разбиваются на элементы-питательные вещества (род *NUTR*) и элементы-материалы (род *MATERIAL*). Элементы-атрибуты разбиваются на элементы — минимальная дневная потребность (род *MIN*), элементы, показывающие сколько каждого питательного вещества имеется в одном фунте матери-

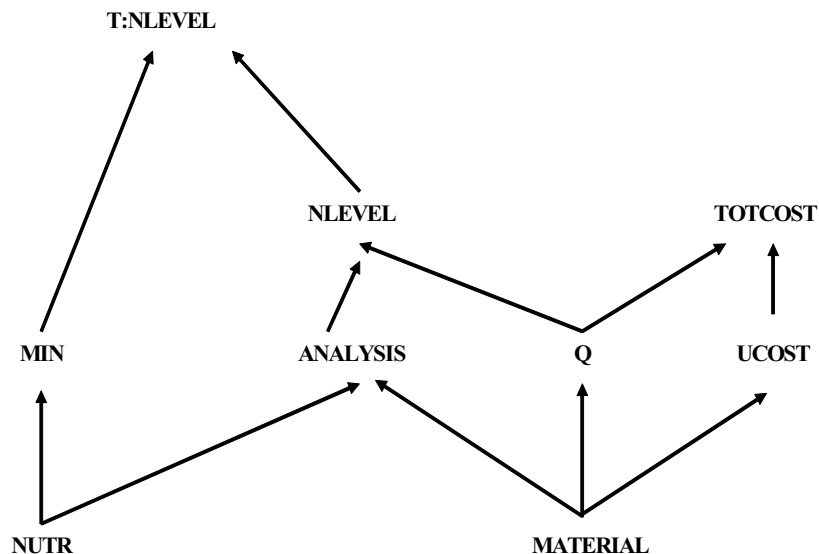


Рис. 2. Родовой граф для модели о рационе питания.

ала (род *ANALYSIS*), элементы-количества (род *Q*), элементы - удельные стоимости (род *UCOST*). Элементы-функции разбиты на элементы, вычисляющие достигнутый уровень питания (род *NLEVEL*) и элемент-полная стоимость (род *TOTCOST*). Наконец, элементы-тесты все собраны вместе в роде (*T : NLEVEL*).

После объединения элементов по соответствующим родам получим родовой граф (рис.2), который не зависит от размерности задачи (так, родовой граф на рис.2 не зависит от количества питательных веществ и материалов).

6.3. Модульное дерево

Среди возможных иерархических вариантов родовой структуры имеется следующий, получаемый путем объединения родов *NUTR* и *MIN* в модуль, интерпретируемый как «данные о питательных веществах» (обозначенный *&NUT_DATA*) и объединения *MATERIAL*, *UCOST* и *ANALYSIS* в другой модуль, интерпретируемый как «данные о материалах» (обозначенный *&MATERIALS*). Полученная модульная структура представлена с помощью модульного дерева на рис. 3. Заметим, что между ребрами родowego графа и ребрами модульного дерева математической связи нет. На рис. 3 также приведен списковая интерпретация дерева.

6.4. Неграфические представления — текстовая схема

Ниже описывается текстовая альтернатива рассмотренным выше графическим моделям структурного моделирования. Здесь для записи родовой и модульной структур задается *текстовая схема*, а элементная структура выражается с помощью элементных детальных таблиц.

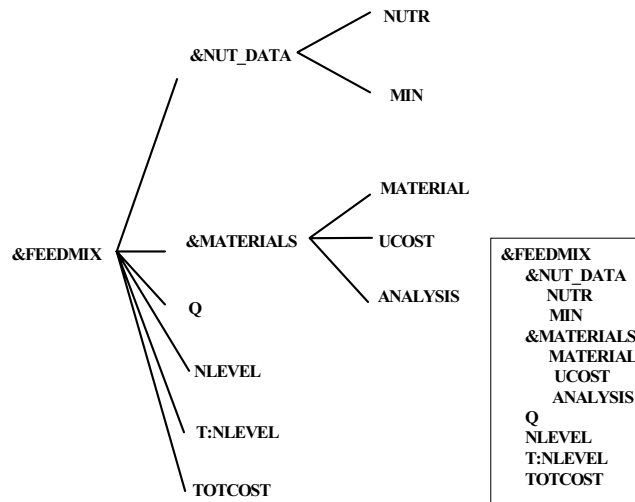


Рис. 3. Модульное дерево (слева) и модульная схема (справа) для модели о рационе питания.

&NUT_DATA NUTRIENT DATA

NUTR_i /pe/ There is a list of NUTRIENTS.

MIN (NUTR_i) /a/ : Real+ For each NUTRIENT there is a MINIMUM DAILY REQUIREMENT (units per day per animal).

&MATERIALS MATERIALS DATA

MATERIAL_m /pe/ There is a list of MATERIALS that can be used for feed.

UCOST (MATERIAL_m) /a/ Each MATERIAL has a UNIT COST (\$ per pound of material).

ANALYSIS (NUTR_i, MATERIAL_m) /a/ : Real+ For each NUTRIENT-MATERIAL combination, there is an ANALYSIS (units of nutrient per pound of material).

Q (MATERIAL_m) /va/ : Real+ The QUANTITY (pounds per day per animal) of each MATERIAL is to be chosen.

NLEVEL (ANALYSIS_i, Q) /f/ ; @SUM_m (ANALYSIS_{im} * Q_m) Once the QUANTITIES are chosen, there is a NUTRITION LEVEL (units per day per animal) for each NUTRIENT calculable from the ANALYSIS.

T:NLEVEL (NLEVEL_i, MIN_i) /t/ ; NLEVEL_i >= MIN_i For each NUTRIENT there is a NUTRITION TEST to determine whether the NUTRITION LEVEL is at least as large as the MINIMUM DAILY REQUIREMENT.

TOTCOST (UCOST, Q) /f/ ; @SUM_m (UCOST_m * Q_m) There is a TOTAL COST (dollars per day per animal) associated with the chosen QUANTITIES.

Рис. 4. Схема для модели о рационе питания.

На рис.4 приведена текстовая схема, выражающая родовую и модульную структуры для модели о рационе питания. Опишем формат и синтаксис, использованный на рис.4:

1. Схема состоит из параграфов, каждый из них занимает строку в модульной схеме и характеризуется одним и тем же отступом. Имеется 2 типа параграфов: модульные параграфы, описывающие модули, начинающиеся всегда с названия модуля, и родовые параграфы, описывающие рода и начинающиеся всегда с названия рода.
2. Название модуля и название рода уникальны и записываются прописными буквами. Названия модулей начинаются с символа & (амперсанд), в то время как название рода начинается с буквы.
3. Каждый параграф состоит из двух частей: формальной части, за которой следует содержательная часть (или интерпретация). Содержательная часть на рис.4 выделена курсивом. Синтаксис содержательной части может быть произвольным, хотя рекомендуется использовать стиль, предложенный в работе Джеффриона, в том числе: использование подчеркнутой уникальной ключевой фразы, записанной прописными буквами, и использование ее ниже записанной прописными буквами. Цель использования описанного стиля состоит в легко читаемой документации. Формальная часть модульного параграфа состоит лишь из имени модуля. Синтаксис формальной части родового параграфа описан ниже.
4. Формальная часть родового параграфа всегда включает индикатор типа (/pe/ (primitive entity — первоначальный объект), /ce/ (compound entity — составной объект), /a/ (attribute — атрибут или параметр, /f/ (function — функция), /t/ (test — тест)).

7. Структурное моделирование и имитационное моделирование

Хотя СМ не предназначалось для имитации, но, тем не менее, ряд статей был посвящен этим аспектам СМ. В работе [5] была исследована возможность использования языка моделирования SML для имитации дискретных событий и было сделано заключение о том, что хотя SML может быть использован для задания статических имитационных моделей, однако он недостаточно хорошо может работать с динамической структурой.

В работе [20] предложены три новых типа элементов, предназначенных для имитационного моделирования дискретных событий: случайные атрибуты (чьи значения генерируются на основе вероятностных распределений), действия (описывающие переходы состояний), и транзакций (использованных для описания сложных событий в терминах последовательности ранее определенных действий

и транзакций). В [21] описан прототип системы управления моделями, основанный на описанных принципах, которые были реализованы в среде базы данных (ORACLE 6.0). Эта схема не зависит от модели, как SML, а фиксирована для всех моделей. В остальном выполняются основные ограничения SML. Система характеризуется тем, что имеется возможность конвертации расширенных структурных моделей в программу на языке SIMSCRIPT II.5.

Подход, предложенный в [29], совершенно другой. Это расширение SML, в котором предлагается начинать схему SML для имитационной модели со стандартного модуля (одного и того же для всех моделей дискретных событий), содержащего первоначальные и составные элементы, определяющие имитацию моделируемой системы. На основе этого подхода может быть реализован процессор, способный к чтению любой такой схемы и генерации имитационной программы на стандартном языке имитационного моделирования типа SIMSCRIPT, который способен описать динамическое поведение моделируемой системы.

8. Заключение

Структурное моделирование — перспективный подход, делающий возможными интеграцию различных парадигм моделирования в исследовании операций [16]. *Перспективными направлениями* дальнейших исследований являются создание программного обеспечения, реализующего СМ [4], [28], [30], имитационное моделирование дискретных событий [21], графовое моделирование [18], интеграция с СУБД [6], объектно-ориентированные системы [27], интеграция моделей [7], [8]), дальнейшее развитие языков задания моделей и манипулирования ими, разработка приложений для ранних и поздних фаз цикла моделирование, не поддерживаемые обычными системами моделирования, разработка СППР [1], [3] на базе СМ [24] и др.

Список цитируемых источников

1. *Никольский В.Н., Рыжаков А.Н., Щербина О.А.* Принципы создания компьютерной системы исследования рекреационных систем на базе математических моделей // *Динамические системы.* — 2005. — Вып. 19. — С. 152–160.
2. *Самарский А.А., Михайлов А.П.* Математическое моделирование. Идеи. Методы. Примеры. — М.: ФИЗМАТЛИТ, 2005. — 320 с.
3. *Сараев А.Д., Щербина О.А.* Системный анализ и современные информационные технологии // *Труды Крымской академии наук.* — Симферополь: СОНАТ. — 2006. — С. 47–59.
4. *Chari K., Sen T.* An Implementation of a Graph-Based Modeling System for Structured Modeling (GBMS/SM) // *Decision Support Systems.* — 1998. — V. 22, N 2. — P. 103–120.
5. *Derrick E.J., Balci O., Nance R.E.* A Comparison Of Selected Conceptual Frameworks For Simulation Modeling // *Simulation Conference Proceedings, 1989.* — P. 711–718.
6. *Dolk D.R.* Model Management and Structured Modeling: The Role of an Information Resource Dictionary System // *Comm. ACM.* — 1988. — V. 31, N 6. — P. 704–718.

7. *Dolk D., Kottemann J.* Model Integration and a Theory of Models // Decision Support Systems. — 1993. — V. 9, N 1. — P. 51–63.
8. *Gagliardi M., Spera C.* Toward a Formal Theory of Model Integration // Annals of Operations Research / I. Maros, G. Mitra (eds.). — v. 58. — P. 405–440.
9. *Geoffrion A.M.* Integrated modeling system // Comp.Sci.Econ.Manag. — 1989. — V. 2. — P. 3–15.
10. *Geoffrion A.M.* An introduction to structured modeling // Management Science. — 1987. — V. 33, N 5. — P. 547–588.
11. *Geoffrion A.M.* The SML language for structured modeling: Levels 1 and 2 // Operations Research. — 1992. — v. 40, N 1. — P. 38–57.
12. *Geoffrion A.M.* The SML language for structured modeling: Levels 3 and 4 // Operations Research. — 1992. — v. 40, N 1. — P. 58–75.
13. *Geoffrion A.M.* The Formal Aspects of Structured Modeling // Operations Research. — 1989. — V. 37, N 1. — P. 30–51.
14. *Geoffrion A.M.* FW/SM: A Prototype Structured Modeling Environment // Management Science. — 1991. — V. 37, N 12. — P. 1513–1538.
15. *Geoffrion A.M.* Structured Modeling: Survey and Future Research Directions // ITORMS. — 1999. — V. 1, N 3.
16. *Hamacher S.* Modeling Systems for Operations Research Problems: Study and Applications, Ph.D. Dissertation. — Paris: Ecole Paris Centrale. — 1995. — 235 pp. (in French).
17. *Harary F., Norman R. Z., Cartwright D.* Structural Models: An Introduction to the Theory of Directed Graphs. — New York: John Wiley & Sons, 1992. — 415 p.
18. *Jones C.V.* Attributed Graphs, Graph-Grammars, and Structured Modeling // Annals of Operations Research. — 1992. — V. 38. — P. 281–324.
19. *Kuhn T.* The structure of scientific revolutions. — Chicago: University of Chicago Press. — 1962.
20. *Lenard M.* Extending the Structured Modeling Framework for Discrete-Event Simulation // Proceedings of the Twenty-Fifth Annual Hawaii International Conference on System Sciences. — Los Alamitos: IEEE Computer Society Press. — 1992. — V. III. — P. 494–503.
21. *Lenard M.* A Prototype Implementation of a Model Management System for Discrete-Event Simulation Models // Proceedings of the 1993 Winter Simulation Conference, IEEE. — Piscataway. — 1993. — P. 560–568.
22. *Manley P. N., Zielinski W. J., Stuart C. M., Keane J. J., Lind A. J., Brown C., Plymale B. L., Napper C.O.* Monitoring ecosystems in the Sierra Nevada: The conceptual model foundations // Environmental Monitoring and Assessment. — 2000. — V. 64. — P. 139–152.
23. *Makowski M.* A structured modeling technology // European J. Oper. Res. — 2005. — V. 166. — P. 615–648.
24. *Maturana S., Eterovic Y.* Vehicle Routing and Production Planning Decision Support Systems: Designing Graphical User Interfaces // International Transactions in Operational Research. — 1995. — V. 2, N 3. — P. 233–247.
25. Modeling Languages in Mathematical Optimization / Ed. J. Kallrath. — Dordrecht: Kluwer Academic Publishers, 2003. — 407 p.

26. *Muhanna W.* On the organization of large shared model bases // *Annals of Operations Research*. — 1992. — V. 38. — P. 359–396.
27. *Muhanna W.* An Object-Oriented Framework for Model Management and DSS Development // *Decision Support Systems*. — 1993. — V. 9, N 2. — P. 217–229.
28. *Neustadter L., Geoffrion A., Maturana S., Tsai Y., Vicuña F.* The Design and Implementation of a Prototype Structured Modeling Environment // *Annals of Operations Research*. — 1992. — V. 38. — P. 453–484.
29. *Pollatschek M.A.* SML for Simulation. — Haifa: Technion. — Faculty of Industrial Engineering and Management. — 1995. — 27 P.
30. *Wright G., Worobetz N.D., Kang M., Mookerjee R., Chandrasekharan R.* OR/SM: A Prototype Integrated Modeling Environment Based on Structured Modeling // *INFORMS J. on Computing*. — 1997. — V. 9, N 2. — P. 134–153.

Получено 21.10.2006