

# СИСТЕМО-ДИНАМИЧЕСКОЕ МОДЕЛИРОВАНИЕ: ПРИНЦИПЫ, СТРУКТУРА И ПЕРЕМЕННЫЕ (на примере Московской области)

С.О. Алехнович, Д.Е. Слизовский,  
Э.Н. Ожиганов

Кафедра политических наук  
Российский университет дружбы народов  
ул. Миклухо-Маклая, 10а, Москва, 117198

В статье освещается методика системно-динамического моделирования политической напряженности на примере Московской области. Авторы подробно излагают принципы, структуру и переменные системно-динамического моделирования.

В современной науке под моделированием понимают изучение моделируемого объекта (оригинала), базирующееся на соответствии определенной части свойств оригинала и замещающего его при исследовании объекта (модели) и включающее в себя построение модели, изучение ее и перенос полученных сведений на моделируемый объект — оригинал.

Эмпирическая морфология политической напряженности, являющаяся объектом моделирования, отражена на следующем рисунке:

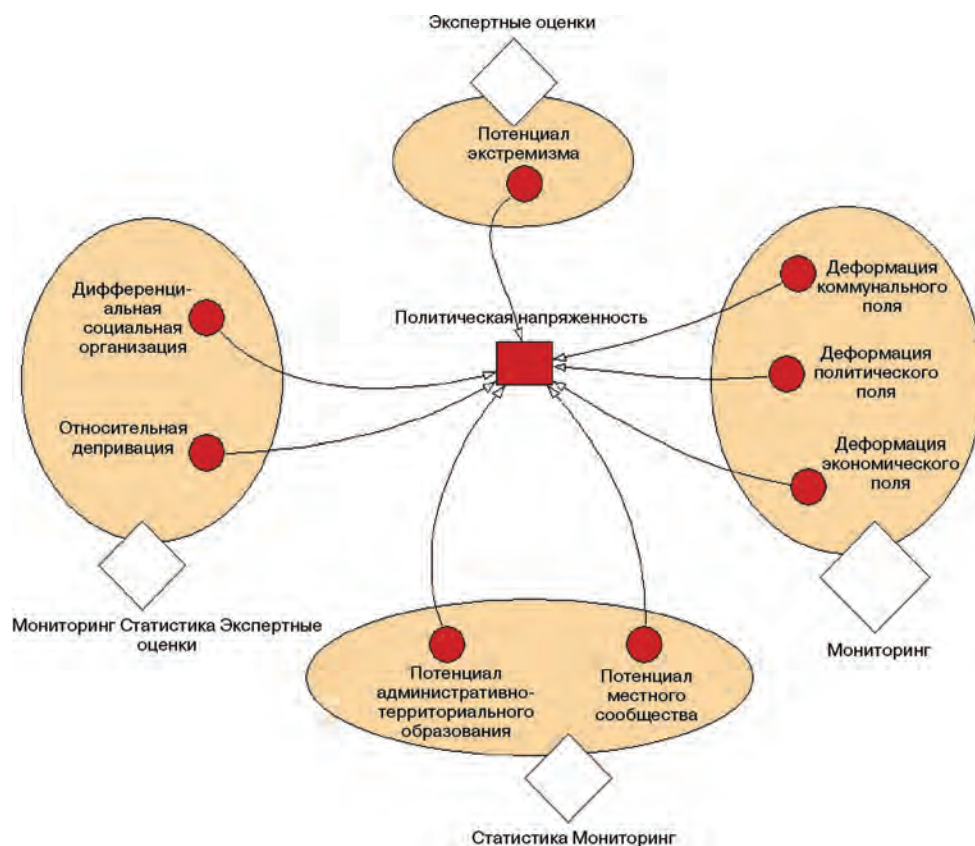


Рис. 1. Факторы политической напряженности

Под динамической системой понимают любой объект или процесс, для которого однозначно определено понятие состояния как совокупности некоторых величин в данный момент времени и задано условие, которое описывает изменение начального состояния с течением времени. Это условие позволяет по начальному состоянию прогнозировать будущее состояние динамической системы. Динамические системы — это механические, физические, химические, биологические, социальные объекты, вычислительные процессы и процессы преобразования информации, совершаемые в соответствии с конкретными алгоритмами.

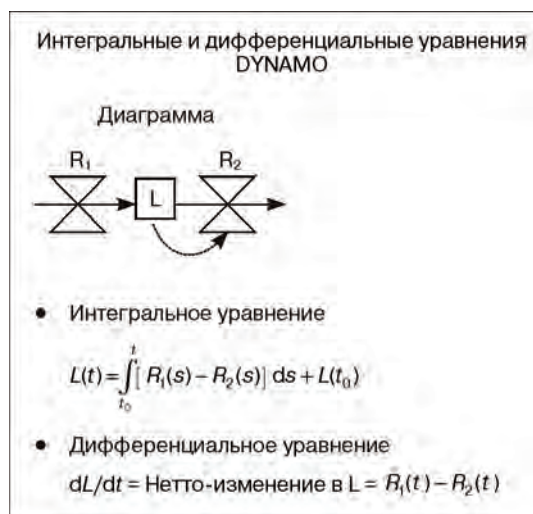
Описания динамических систем также разнообразны: с помощью дифференциальных уравнений, дискретных отображений, теории графов, теории марковских цепей и т.д. Выбор одного из способов описания задает конкретный вид математической модели соответствующей динамической системы. Например, в среде пакетов *Powersim Studio*, *Vensim*, *Stella* и им подобных программ консолидированная математическая модель реализуется средствами визуального программирования.

**Методология системной динамики** базируется на предположении, что поведение сложных объектов главным образом определяется их информационно-логической структурой. Она отражает не только физические и технологические аспекты социальных, производственных и других процессов, но и политику и традиции, которые явно или неявно определяют процесс принятия решений. Такая структурная схема содержит источники усиления, временных задержек и информационных обратных связей (например, подобных тем, которые встречаются в сложных инженерных системах).

Второй аспект методологии системной динамики — направленность на потоковую структуру — заключается в предположении, что сложные организации более эффективно представляются в терминах лежащих в их основе потоков, нежели в терминах отдельных функций.

Третий аспект — использование и модификация известных методов представления потоковых диаграмм, математического и имитационного моделирования. Для системно-динамических проектов созданы формальные потоковые диаграммы, представляемые в виде систем дифференциальных уравнений. Как потоковые диаграммы, так и системы уравнений выражают управленческие связи с помощью двух ключевых категорий: «накопителей» и «потоков». «Накопители» представляют собой такие объекты реального мира, в которых сосредотачиваются некоторые ресурсы: знания (идеи), фонды, источники и т.п. «Потоки» — это все активные компоненты системы: потоки усилий (попыток), информационные потоки, расходные платежи и т.п.

Если система представима в виде сети «накопителей» и «потоков», то соответствующая системно-динамическая модель может быть реализована в виде компьютерной программы. С помощью такой программы можно провести экспериментальное тестирование предлагаемых изменений управленческой политики. С помощью языка моделирования (например, компилятора *DYNAMO*) можно эффективно решать системы линейных и нелинейных алгебраических и дифференциальных уравнений, содержащих до нескольких тысяч переменных, при этом от пользователя не требуется глубоких знаний программирования.



**Рис. 2.** Уравнения компилятора *DYNAMO*

В системно-динамическом моделировании моделируемый объект отображается в виде динамической системы, состоящей из «накопителей» (уровней, фондов), связанных между собой управляемыми «потоками». Количественно каждый «накопитель» описывается уровнем его содержимого, а каждый «поток» — темпом (скоростью) перемещения. Темпы перемещения вычисляются на основе информации об уровнях содержимого «уровней». Таким образом, моделируемый объект представляется в виде информационной системы с обратной связью.

«Накопитель» описывает величины, непрерывные по диапазону своих значений и дискретные во времени — их можно определить как переменные состояния системы, значения которых формируются за счет накопления разностей между входящими и выходящими потоками.

«Потоки» — аналоги процессов преобразования накоплений в системе — они перемещают содержимое «накопителей» и отражают либо материальные, либо информационные процессы. Их интенсивность (темп) определяется управленческими решениями, которые формируются на основании информации о состоянии уровней. Функции решений (или уравнения темпов) представляют собой формализацию правил, определяющих, каким образом информация об уровнях приводит к выбору текущих значений темпов потоков.

Модели «потокового» типа относят к динамическим моделям с обратными связями. Так как реальные системы обладают инерционностью, в их структуре имеются элементы, определяющие запаздывания передачи изменения по контуру обратной связи.

В каждой модели системной динамики присутствует особая дискретная переменная — время, выбор единицы измерения которой (временного шага моделирования), как и интервала моделирования, осуществляется разработчиком модели.

В математическом смысле модель системной динамики представляет собой систему конечно-разностных уравнений, решаемую на основе численного алгоритма интегрирования по схеме *Эйлера* или *Рунге-Кутты* с заданными начальными значениями уровней.

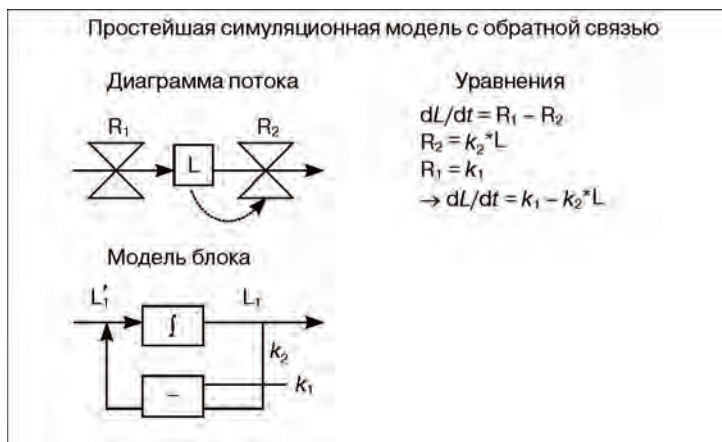


Рис. 3. Блок системной динамики

Основные принципы системной динамики могут быть сформулированы следующим образом:

- поведение системы — это следствие проявления ее структуры и взаимодействия элементов;
- структура системы и характер взаимосвязей между элементами системы, определяющие ее поведение, более важны для понимания поведения системы, нежели количественные оценки;
- состояние системы и ее структура есть причина изменений, а не их результат;
- изучить систему — значит определить ее структуру и установить отношения между ее элементами;
- определяющее значение в поведении системы имеет взаимодействие контуров обратной связи в ее структуре;
- уровни и темпы являются необходимыми и достаточными переменными для описания любой динамической системы;
- при построении системно-динамических моделей следует опираться на принцип «непосредственной верификации» (прямую проверку утверждений, формулирующих данные наблюдения и эксперимента);
- при изучении системы важно концентрировать внимание на анализе ответственности «политик», а не на получении точных количественных оценок.

Имитационное исследование характеризуется внутренней логикой, которая выражается в последовательности ряда этапов и позволяет сочетать особенности экспериментального подхода и специфику использования средств компьютерной поддержки.

Имитационное исследование носит последовательно-итеративный характер, связанный, с одной стороны, с четким порядком смены стадий и этапов, а с другой, — с цикличностью выполнения отдельных этапов (в рамках отдельных стадий и между стадиями).

При разработке моделей используется визуальное программирование. Модель, включающая в себя различные элементы, строится на понятийном уровне.

Разработчик располагает на экране элементы, создает связи между ними, вводит различные зависимости, создает временную динамику развития системы. В модель вводятся различные управляющие элементы.

Система имеет развитые способы представления результатов моделирования: временные графики, таблицы. Вид результатов может быть легко приведен к требуемому стандарту.

Диаграмма причинно-следственных связей показывает, в каких отношениях находятся между собой переменные модели, и представляет собой размеченный граф, элементами которого являются переменные, представленные своими идентификаторами или полными именами, а стрелки идут от переменных-причин к переменным-следствиям.

При построении системно-динамической модели необходимо понимать, что:

— петля обратной связи — базовый (основной) структурный элемент любой системы;

— «накопитель» и «поток» (темп) являются основой для петлевой структуры;

— уровни являются аккумулярующим (интегрирующим) компонентом (например, население как интегральный показатель от темпов рождения и миграции);

— уровни изменяются только за счет «потоков» (темпов);

— все уровни существуют в сохраняемых подсистемах и содержат сохраняемые данные;

— «потоки» (темпы) зависят только от «накопителей» и «констант»;

— решения находятся всегда внутри петель обратной связи;

— каждое уравнение должно иметь количественную размерность.

В соответствии с классификацией переменных и причинно-следственных связей строится диаграмма потоков, в которой отражаются константы и порядок вычисления независимых переменных. Диаграмма потоков акцентирует внимание на физической структуре и отображает накопления (уровни, фонды) материалов и информации при их перемещении в рамках системы.

При построении диаграммы потоков приняты стандартные графические обозначения для всех типов переменных. «Накопители» помещаются в прямоугольную рамку, «поток» (темп) изображается в виде стрелки с регулятором, дополнительные переменные — «конверторы» — могут помещаться в круг, «константы» — в ромб, и отображаться своими идентификаторами (полными именами).

Формулируя иначе, модель есть целенаправленное представление исследуемого объекта, реальное или воображаемое. Имитационная модель есть модель, которая воспроизводит поведение объекта за определенный период времени, так что в этом смысле имитационная модель является динамической. Значения всех переменных, входящих в имитационную модель, вычисляются в каждый момент модельного времени. Затем, через определенный интервал, на основе старых значений вычисляются новые значения переменных и т.д. Таким образом, имитационная модель развивается по определенной траектории в течение заданного отрезка модельного времени.

Для построения имитационных моделей динамических систем используются переменные четырех типов: «накопитель», «поток», «конвертор» и «время».

Переменная «накопитель» равна объему (количеству) некоторого «продукта», материального или информационного, накопленного в некотором хранилище за время «жизни» модели с начального по текущий момент. «Продукт» может поступать в «накопитель» или извлекаться из него, поэтому значение «накопителя» в текущий момент времени можно вычислить как сумму его значения в предыдущий момент и величины, равной разности величин входящего и исходящего потоков продукта за единицу модельного времени.

Переменные этого типа могут характеризовать объемы накопленной информации, служить оценкой субъективных вероятностей наступления некоторых событий к определенному моменту времени, выражать меру влияния одних субъектов некоторого процесса на других. Средние величины всех видов также можно рассматривать как информационные «накопители» специального вида.

Переменная типа «поток» равна объему (количеству) «продукта», который поступает или извлекается из соответствующего «накопителя» в единицу модельного времени. Значение этой переменной может изменяться в зависимости от внешних воздействий на нее. В частности, поток можно представить как функцию от значений других потоков и «накопителей». Простейший пример цикла с обратной связью образует входящий поток, величина которого зависит от значения «накопителя», в который этот поток поступает.

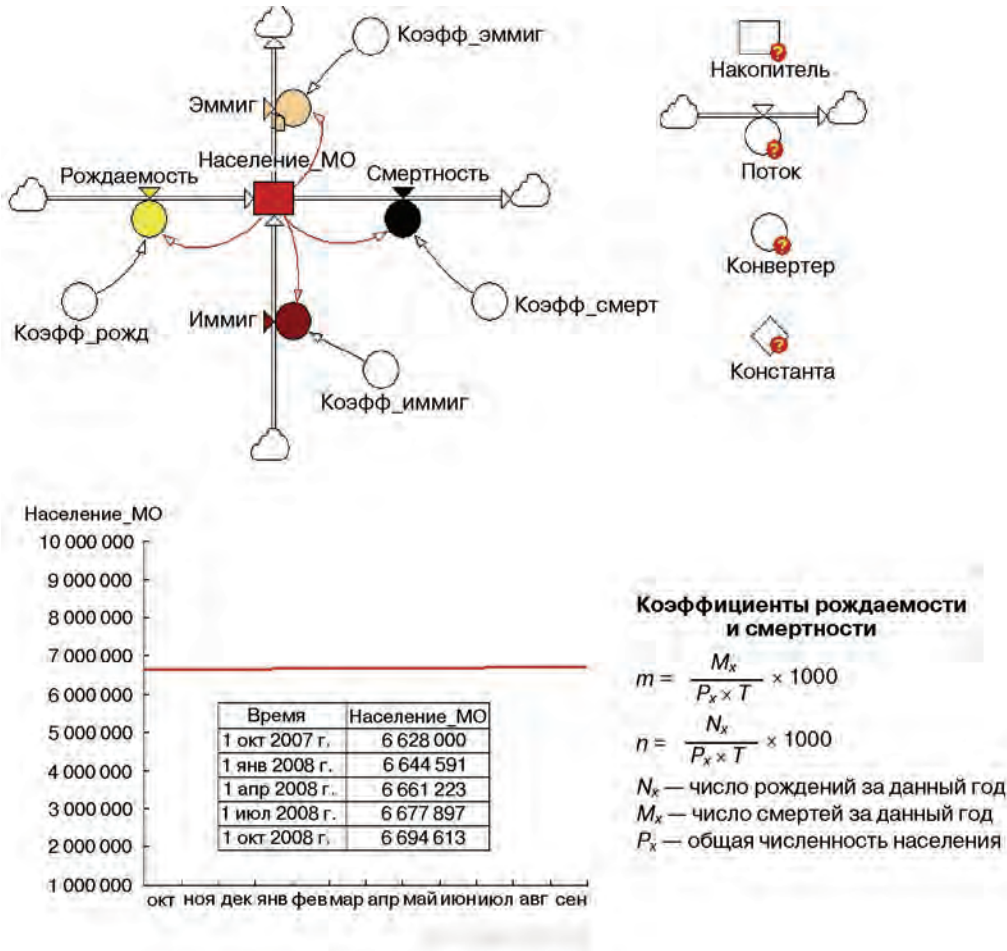
«Накопители» характеризуют статическое состояние системы, а «потоки» — ее динамику. Если, например, представить себе, что в какой-то момент времени все процессы в системе останутся, то «накопители» будут иметь те значения, которые были на момент остановки, а «потоки» будут равны нулю. С другой стороны, о величине «потока» можно судить только за определенный промежуток времени.

Помимо «накопителей» и «потоков», при построении имитационных моделей динамических систем используются вспомогательные переменные, которые называют «конверторами». Эти переменные могут быть равны константам или значениям математических функций от других переменных (в том числе и от переменной «время»), т.е. позволяют преобразовывать («конвертировать») одни числовые значения в другие.

Переменная «время» является первичной для имитационной модели динамической системы: ее значение генерируется системным таймером и изменяется дискретно, т.е., начиная с некоторого начального значения, время за каждый такт увеличивается на заранее заданную величину, которая служит единицей модельного времени. Число тактов и единица времени являются параметрами «прогона» модели и определяются разработчиком модели.

На рис. 4 приводится простейший пример, описывающей динамику населения Московской области в течение года в зависимости от четырех процессов: рождаемости, смертности, иммиграции и эмиграции.

«Накопитель» в данной графической нотации — население Московской области с начальным значением 6 628 000 человек на 1 октября 2007 года. «Потоки» отображают показатели рождаемости, смертности, иммиграции и эмиграции для соответствующего периода времени, «конвертеры» содержат значения соответствующих коэффициентов (рис. 5).



**Рис. 4.** Модель динамики населения

Name	Dimensions	Unit	Definition
Иммиг		yr <sup>-1</sup>	Кoeff_иммиг*Население_МО
Кoeff_иммиг		yr <sup>-1</sup>	0,105 <<1/yr>>
Кoeff_рожд		yr <sup>-1</sup>	0,091 <<1/yr>>
Кoeff_смерт		yr <sup>-1</sup>	0,171 <<1/yr>>
Кoeff_эммиг		yr <sup>-1</sup>	0,015 <<1/yr>>
Население_МО			6628000
Иммиг.in			Иммиг
Рождаемос...			Рождаемость
Смертност...			Смертность
Эммиг.out			Эммиг
Рождаемость		yr <sup>-1</sup>	Кoeff_рожд*Население_МО
Смертность		yr <sup>-1</sup>	Кoeff_смерт*Население_МО
Эммиг		yr <sup>-1</sup>	Население_МО*Кoeff_эммиг

**Рис. 5.** Значения коэффициентов модели динамики населения



Динамика изменения численности населения Московской области при заданных переменных и их значений представлена на графике, где по горизонтальной оси отложен период модельного времени, а по вертикальной оси — численность населения Московской области. Результаты имитационного моделирования показывают, что, при заданных условиях, к октябрю 2008 года население Московской области увеличится на 66 613 человек и его численность составит 6 694 613 человек.

С точки зрения представляемой методологии, изучение процессов в системе по отдельности невозможно — только изучение динамики всей системы со всеми ее связями дает корректное понимание исследуемой проблемы. Все системы, какими бы сложными они не были, построены на замкнутых циклах обратной связи зависимостей переменных в структуре имитационной модели.

Все показатели можно разделить на два основных вида: абсолютные и относительные. Абсолютные показатели (или величины) — это просто суммы событий (явлений) на момент времени или в интервале времени (например, за год).

Абсолютные показатели сами по себе неинформативны, поэтому они используются в аналитической работе обычно лишь как исходные данные (сырье) для расчета относительных показателей. Они непригодны для сравнительного анализа, потому что их величина зависит от величины, с которой они всегда находятся в определенной пропорции, или иначе можно сказать — которая их продуцирует.

В сравнительном анализе (для сравнений любого рода, в статике или в динамике) используют только относительные показатели. Относительными они называются потому, что всегда представляют собой дробь, отношение к той величине, которая их продуцирует, и таким образом различие в величинах устраняется. Главное требование любого сравнения каких-либо двух (или нескольких) признаков — уравнивать все прочие признаки изучаемого явления, кроме тех, которые непосредственно сравниваются. Только тогда можно получить представление о действительной разнице между изучаемыми признаками.

Приведение изучаемых явлений к сопоставимому виду, элиминирование всех посторонних для данного сравнения факторов — сложная задача из-за трудностей выделения объекта наблюдения в «чистом виде» из общей массы общественных явлений. Ее решают, как правило, лишь с помощью абстрагирования на экспертном уровне, и в этом кроется определенная опасность неадекватного представления об изучаемом явлении.

Все объекты исследований могут быть разделены на три группы: 1) **размерные** — изменяющиеся как при изменении единиц измерения, так и при преобразованиях систем отсчета; 2) **константы** — изменяющиеся при перемене единиц измерения, но не чувствительные к преобразованиям систем отсчета и 3) **безразмерные** — не реагирующие ни на смену системы отсчета, ни на смену единиц измерения. Размерные (измеримые) и безразмерные (исчислимые) величины приобретают свой смысл в сравнении: размерные — в сравнении с определенным эталоном, исчислимые — в сравнении с единицей.



Сформулируем ряд понятий, связанных с процессом нормализации:

— адекватность математической модели — соответствие математической модели экспериментальным данным по выбранному критерию;

— эксперимент — система операций, воздействий и (или) наблюдений, направленных на получение информации об объекте при исследовательских испытаниях; активный эксперимент — эксперимент, в котором уровни факторов в каждом опыте задаются исследователем;

— фактор — переменная величина, по предположению влияющая на результаты эксперимента; в большинстве моделей, используемых в планировании эксперимента, предполагается, что факторы могут рассматриваться как детерминированные переменные. Обычно факторы выражаются в безразмерных единицах масштаба и обозначаются буквами  $x_i$ ,  $i = 1, 2, \dots, k$ ;

— уровень фактора — фиксированное значение фактора относительно начала отсчета;

— основной уровень фактора — натуральное значение фактора, соответствующее нулю в безразмерной шкале;

— нормализация факторов — преобразование натуральных значений факторов в безразмерные; за единицу масштаба безразмерной системы координат принимается некоторый интервал в натуральных единицах.

Приведем пример т.н. векторной оптимизации, где локальные критерии оптимальности имеют различные единицы измерения, а в основу нормализации критериев кладется понятие «идеального вектора», то есть вектора с «идеальными» значениями параметров:

$$F = \{f_1, f_2, \dots, f_k\}.$$

В нормализованном пространстве критериев вместо действительного значения критерия  $f_q$  рассматривается безразмерная величина:

$$f_q^n = \frac{f_q}{f_{.q}^n}, \quad q = \overline{1, k}.$$

Успешное решение проблемы нормализации зависит от того, насколько правильно и объективно удастся определить идеальные значения.

Способ выбора идеального вектора  $f^n$  и определяет способ нормализации.

Основные способы нормализации могут быть представлены следующим образом:

1) идеальный вектор определяется заданными величинами критериев:

$$F^n = F^{\text{зад}} = X \{f_q^{\text{зад}}\}, \quad q = \overline{1, k}.$$

Недостатком этого способа является сложность и субъективность назначения  $F^{\text{зад}}$ ,

2) в качестве идеального вектора выбирают вектор, параметрами которого являются максимально возможные значения локальных критериев:

$$F^n = F_{\max} = \{f_{1 \max}, f_{2 \max}, \dots, f_{k \max}\}.$$

Недостатком этого способа является то, что он существенно зависит от максимально возможного уровня локальных критериев. В результате равноправие критериев нарушается и предпочтение отдается варианту с наибольшим значением локального критерия;

3) в качестве параметров идеального вектора принимают максимально возможный разброс соответствующих локальных критериев, то есть:

$$f_q^H = f_{q \max} - f_{q \min}, \quad q = \overline{1, k}.$$

При нормализации фактора наряду с изменениями масштаба изменяется начало отсчета. Значение  $i$ -го фактора в безразмерной системе связано со значением этого фактора  $x_i^H$  в натуральной системе (в именованных единицах) формулой:

$$x_i = \frac{x_i^H - x_{0i}^H}{\Delta x_i^H},$$

где  $x_{0i}^H$  — основной уровень фактора, принимаемый за начало отсчета;  $\Delta x_i^H$  — интервал в натуральных единицах масштаба, соответствующий одной единице масштаба в безразмерных переменных.

В примере с моделью динамики населения Московской области натуральными единицами являются коэффициенты рождаемости и смертности, вычисляемые по следующим формулам:

$$m = \frac{M_x}{P_x \times T} \times 1000;$$

$$n = \frac{N_x}{P_x \times T} \times 1000,$$

где  $N_x$  — число рождений за данный год;  $M_x$  — число смертей за данный год;  $P_x$  — общая численность населения.

**Общие коэффициенты естественного движения населения (на 1000 человек)**

Московская область	2002 г.	2003 г.	2004 г.	2005 г.	2006 г.
Число родившихся					
	8,2	8,8	9,1	9,0	9,1
Число умерших					
	17,9	17,9	17,5	17,5	17,0

Для использования этих коэффициентов в имитационной модели динамики населения их необходимо нормализовать, преобразовав в т.н. безразмерные. Поскольку натуральными единицами заданы коэффициенты рождаемости (9,1) и смертности (17,0), то в качестве критериев нормализации принимаются их максимально возможные значения, т.е. 100, а константой является 1 (единица).

Следовательно, нормализованный коэффициент рождаемости для модели — 0,091, а нормализованный коэффициент смертности — 0,171.

Тот же алгоритм применяется и в том случае, когда натуральными единицами заданы проценты ответов на вопросы анкеты, где максимально возможные

значения составляют 100%. То есть если на определенный вопрос ответили определенным образом 25% населения муниципального образования, то нормализованный показатель будет иметь значение 0,25.

В плане применения математики цель программной среды имитационного моделирования как раз состоит в том, чтобы помочь исследователю решить проблемы динамики исследуемых процессов, с математическим выражением которых было бы трудно иметь дело без помощи среды моделирования. Программы имитационного моделирования в значительной степени освобождают аналитика от необходимости обращаться к лежащей в их основе математике и от деталей спецификации языка программирования. Иначе говоря, графическая нотация позволяет строить и использовать модели, не отвлекаясь на вопросы о том, какова математика модели или как работает язык моделирования. Однако для объяснения методологии возникает потребность осветить математические принципы динамики систем немного глубже.

Язык моделирования позволяет дать простое представление сложных динамических систем. Прежде всего, следует помнить, что динамику системы определяют ее структурные элементы (*levels*). Следующие уравнения показывают основную математическую форму языка моделирования:

$$levels_t = \int_0^T rates_t dt \quad \text{или} \quad \frac{d}{dt} levels_t = rates_t; \quad (1)$$

$$rates_t = \mathbf{g}(levels_t, aux_t, data_t, const); \quad (2)$$

$$aux_t = \mathbf{f}(levels_t, aux_t, data_t, const); \quad (3)$$

$$levels_0 = \mathbf{h}(levels_0, aux_0, data_0, const). \quad (4)$$

В этих уравнениях  $\mathbf{g}$ ,  $\mathbf{h}$ ,  $\mathbf{f}$  — произвольные, нелинейные, потенциально изменяемые во времени векторнозначные функции. Уравнения (1) представляют эволюцию системы в течение определенного времени, уравнение (2) — вычисление показателей (*rates*), детерминирующих эту эволюцию, уравнение (3) — промежуточные результаты, необходимые для вычисления показателей, и уравнение (4) — инициализацию системы.

Уравнения (1) написаны с использованием как интегральной, так и дифференциальной нотации. Оба эти уравнения имеют одно и то же значение.

Символы *aux*, *const*, *data*, *levels* и *rates* представляют различные типы переменных:

— *aux<sub>t</sub>* — «вспомогательные» элементы. Переменные этого типа вычисляются из *levels*, *constants*, *data* других *auxiliaries*. У вспомогательных переменных нет накопителей значения, и их текущие значения независимы от значений переменных в предыдущих итерациях;

— *const* — «константы». Переменные данного типа не изменяются со временем;

— *data<sub>t</sub>* — «данные», которые имеют значения, изменяемые в течение долгого времени, но независимые от процессов, воздействующих на другие переменные;

—  $levels_t$  — «уровни» (также называемые «фондами» или «состояниями»). Они изменяются только в течение определенного времени, а их значения всегда зависят от значений, которые они (и другие переменные) имели в предыдущих итерациях. Уравнение (1) показывает, как уровни интегрируются или «накапливаются» на основе своих значений и значений других переменных в системе. Данный тип переменных в конечном счете определяет динамическое поведение системы;

—  $rates_t$  — «показатели» (также называемые «потоками») — переменные, которые непосредственно изменяют уровни (фонды). Показатели по существу являются такими же переменными, как и вспомогательные элементы (*auxiliaries*), и отличаются только по способу, которым они используются в модели.

Уравнение (4) показывает, что значение уровней и вспомогательных элементов зависят друг от друга в любое время. Уравнение (3) показывает, что значения вспомогательных элементов всегда зависят от других вспомогательных элементов. Оба из этих уравнений должны удовлетворять дополнительному условию о том, что значение любого элемента в конкретное время не может зависеть от самого себя, но в предыдущих итерациях значение переменной может, однако, зависеть от своих параметров. Модель, в которой значение переменной в конкретное время может быть определено независимо от параметров этой переменной, считается причинной (каузальной) моделью.

Пока модель является причинной, программа имитационного моделирования может определить значения всех переменных и использовать причинную структуру для определения соответствующей последовательности вычислений. Порядок, в котором пользователь определяет переменные, не имеет никакого значения к вычислительной последовательности. Например, программа *Powersim Studio* выполняет следующие шаги в симуляции модели.

#### 1. Подготовка.

Все константы приводятся в соответствие с определенными для них значениями, и обрабатываются любые изменения значений констант в файлах или ссылках на таблицы данных для констант. Все данные (как внешние переменные и данные, необходимые для вычисления результатов) читаются, значения загружаются, и вычисляются уравнения данных (в том числе данные из таблиц). Вычисляется значение начального времени, и все данные (*data*) устанавливаются в соответствии с их значениями в это время. В конце подготовки все константы (*constants*) и данные (*data*) будут иметь приписанные значения.

#### 2. Инициализация.

Для каждого уровня (фонда) в модели *Powersim Studio* определяет начальное значение для того уровня. В конце инициализации у всех уровней в модели будут определены значения. В процессе вычисления этих значений некоторым вспомогательным элементам (*auxiliaries*) также могут быть приписаны определенные значения.

#### 3. Сохранение результатов (для дифференциальных уравнений).

Записанные результаты — это значения уровней (в специфическое время) и вспомогательных элементов, которые определили состояние уровней.

4. Вычисление вспомогательных элементов (*auxiliaries*).

Для каждого из вспомогательных элементов в модели вычисляется значение в соответствии с его уравнением. Этот процесс является рекурсивным и заканчивается, когда переменными, которые используются, будут являться константы, данные и уровни. По очевидным причинам этот порядок вычисления всегда реализуется перед актуальным имитационным моделированием.

5. Вычисление реальных показателей (*net rates*).

На основе вспомогательных элементов (*auxiliaries*) вычисляется реальное значение изменений для каждого уровня (фонда) в имитационной модели.

## 6. Интеграция.

Для интеграции по методу *Эйлера* результаты шага 5 мультиплицируются с помощью параметра *TIME STEP* (временной шаг) и добавляются к значениям уровней (фондов). Для интеграции по методу *Рунге-Кутты* шаги 4 и 5 повторяются неоднократно (но во время данного вычисления информация не сохраняется). В конце этого шага *TIME* (*Время*) продвигается с помощью параметра *TIME STEP* и вычисляются значения уровней для нового времени.

Согласно стандартной практике метод динамики систем (*SD*) агрегирует дискретные процессы, представляя их в форме непрерывных потоков. Технология динамики систем была первоначально разработана для тех систем, состояния которых изменяются в непрерывном виде в течение определенного времени. Модель *SD* типично агрегирует дискретные события основной системы на уровне, где события могут быть рассмотрены как части непрерывного потока. К примеру, проведение отдельных мероприятий может рассматриваться как поток подобных мероприятий в течение определенного времени.

Алгоритм непрерывной симуляции использует интеграцию с учетом фактора времени, что дает возможность представить процесс развития систем. Например, в дополнение к интеграции программа *Powersim Studio* поддерживает и алгоритм «накопления». Для сравнения, в моделях электронных таблиц используется исключительно алгоритм накопления, поскольку они не могут интегрировать данные. «Накопление» означает, что значения могут быть добавлены или вычтены из некоторого состояния переменных.

*Powersim Studio* поддерживает три вида потоков, а именно интеграцию, накопление и логические операции. Интеграция поддерживается только для непрерывных типов данных, то есть действительных чисел (*real numbers*) и комплексных чисел (*complex number*). Накопление поддерживается только для цифровых типов данных, то есть реальных, комплексных и целых. Логические операции поддерживают только логические типы данных. Выбор типа потока определяется типом вовлеченных переменных и порядком интеграции для этих переменных.

Во время моделирования непрерывной системы уровни (фонды) вычисляются с помощью числовой интеграции в течение определенного времени. Уровень  $L$  с чистым потоком (*net flow*)  $F$  представляет следующее развитие в течение некоторого времени:

$$L_t = L_0 + \int_0^t F dt.$$

Симуляция продвигается на один временной шаг некоторого времени, таким образом, что интеграция разделена на шаги, а результат одного шага используется для вычисления следующего шага. К примеру, значение уровня  $L$  во время  $t + \Delta t$  равно значению во время  $t$  плюс интеграл потока, идущего от  $t$  к  $t + \Delta t$ , где  $\Delta t$  — временной шаг моделирования:

$$L_{t+\Delta t} = L_t + \int_t^{t+\Delta t} F dt.$$

Интеграция потока по одному временному шагу аппроксимируется согласно параметрам или установкам интеграции симуляционной модели. Самая простая установка — интеграция первого порядка метода Эйлера, с фиксированным временным шагом. Более точные результаты моделирования могут быть достигнуты при использовании одного из методов интеграции более высокого порядка, таких как метод Рунге-Кутты (хотя это замедлит симуляцию, поскольку потребуются больше вычислений). Временной шаг может быть фиксированным или изменяемым. Если временной шаг будет изменяемым, то процесс интеграции подразделит интервал интеграции далее с целью удержать ошибку интеграции в определенных рамках, определенных самим пользователем.

Интеграция только возможна на непрерывных типах данных, таких как действительные числа. Но это только одна из причин для использования накопления в дополнение к интеграции в языке симуляции, поскольку есть множество ситуаций, где возникает необходимость смоделировать транзакции, которые имеют место как события в определенный момент времени.

Накопление может быть выражено весьма легко в форме суммы:

$$L_t = L_0 + \sum_0^{t-\Delta t} F.$$

Что касается числовой интеграции, процесс накопления разделен на временные шаги, где ценность одного временного шага используется, чтобы найти значение для следующего шага:

$$L_{t+\Delta t} = L_t + F_t.$$

При накоплении единица измерения является одной и той же для потока и уровня (фонда), поскольку мы не умножаем поток с каждым временным шагом. Кроме того, накопление определяется как для дискретных, так и непрерывных чисел. Это означает, что мы можем теперь представить целое число в дополнение к реальным и комплексным уровням (фондам).

Во время моделирования для модельной переменной создается временной ряд, значения которого могут непрерывно передаваться для связанной переменной набора данных. Если набор данных установлен в автоматическом режиме, то этот процесс завершится в конце моделирования.

Например, набор данных электронной таблицы позволяет передавать данные между *Powersim Studio* и *Microsoft Excel*. Набор данных поддерживает двустороннюю передачу, осуществляя экспорт значений определенных перемен-

ных в электронную таблицу, в то же самое время импортируя значения для других переменных. Аналитик может соединить набор данных с уже существующей электронной таблицей на его компьютере или разрешить набору данных создать новую электронную таблицу, связанную с тем или иным компонентом модели.

**SYSTEM-DYNAMICAL MODELING:  
THE PRINCIPLES, STRUCTURE AND VARIABLES  
(on the example of the Moscow region)**

**S.O. Alekhnovich, D.E. Slizovskiy,  
E.N. Ozhiganov**

The Department of Political Science  
Russian People's Friendship University  
*Mikluho-Maklaya str., 10a, Moscow, Russia, 117198*

The given article analyses the methods of system-dynamical modeling of political tension on the example of the Moscow region. The authors provide detailed principles, structure and variables of system-dynamical modeling.