

# Программные инструменты верификации моделей сложного взаимодействия факторов в экономике

В. Б. Гусев, П. В. Куракин

Институт проблем управления им. В. А. Трапезникова РАН, г. Москва, Россия

**Аннотация.** В работе предложен подход к разработке человеко-машинной технологии верификации моделей сложного взаимодействия факторов в экономике. Аргументируется, что задача такой верификации требует качественно новых по сравнению с большинством традиционных задач численного моделирования подходов и концепций. Описано программное обеспечение, частично уже реализованное, призванное ответить на существующий вызов. В основу решения кладется визуальное проектирование экземпляра задачи и его модификаций в среде специализированного графического редактора.

**Ключевые слова:** экономика, взаимодействие факторов, взаимно дополнительные факторы, взаимно компенсирующие факторы, верификация, графический редактор, вычислительный эксперимент.

DOI 10.14357/20718632190106

## Введение

Экспертные модели предназначены для оценки долгосрочных последствий взаимодействия факторов хозяйственной деятельности на крупномасштабном уровне (территории, региона, государства). Исходный этап анализа – подготовка экспертных данных, представляющих первичные оценки степени взаимовлияния рассматриваемых факторов друг на друга. Результат анализа – полные оценки степени взаимовлияний факторов, учитывающие расширенный набор цепочек трансакций. Применяемые в модели оценки представляются в баллах и интерпретируются как значения истинности в многозначной логике. Для операций над оценками используются соответствующие действия многозначной логики [1]. Отличительной чертой экспертного подхода является включение в анализ набора факторов, достаточно полно

определяющих причинно-следственные связи рассматриваемого объекта. Используемые данные основываются на мнении экспертов и могут не иметь отражения в официальной статистике, дополняя их для целей анализа. Полные оценки степени взаимовлияний факторов можно использовать при сравнении различных сценариев развития с целью выбора управляющих воздействий при долгосрочном планировании.

Действие одних факторов на другие затрагивает различные аспекты. В результате сложения эффекта от всех цепочек воздействий, исходящих от каждого фактора-причины и заканчивающихся факторами-следствиями, формируется системный эффект, определяемый полной совокупностью возникающих косвенных связей.

При принятии решений можно выбирать интерпретацию применяемых числовых данных как оценки интенсивности взаимодействия факторов. Конкретный аспект взаимодействия

обладает своей спецификой и требует соответствующего математического аппарата [1].

Отличительной чертой экспертного подхода является включение в анализ набора факторов, достаточно полно определяющих причинно-следственные связи рассматриваемого объекта. Используемые данные основываются на мнении экспертов и могут не иметь отражения в официальной статистике, дополняя их для целей анализа. Получаемые в результате анализа полные оценки степени взаимовлияний факторов можно использовать при сравнении различных сценариев развития с целью выбора управляющих воздействий при долгосрочном планировании.

Схема «примитивных» (исходных, базовых) взаимодействий факторов представляется экспертной матрицей  $A$ . Коэффициент  $a_{ij}$  этой матрицы означает оценку сверху первичного прироста фактора  $i$ , вызываемого приращением фактора  $j$  (аналог пропускной способности канала, направленного от фактора  $j$  к фактору  $i$ ). Задачей анализа транзитивного замыкания взаимодействий из рассматриваемой совокупности факторов является расчет оценок полного взаимовлияния факторов, учитывающих как прямые, так и косвенные воздействия, а также получение выводов об эффективности механизмов управления с точки зрения их воздействия на целевые показатели с индексами  $i \in C$ .

Применяемые в модели оценки представляются в баллах и интерпретируются как значения истинности в многозначной логике. Для операций над оценками используются соответствующие действия многозначной логики, обладающей следующими свойствами:

- последовательное применение в цепочке 2-х операций (логическое умножение операндов) с противоположными по знаку оценками дает отрицательную оценку влияния;
- если связь не входит ни в один цикл, то в результате рефлексии ее оценка должна сохранять свое значение;
- результат параллельного действия связей (логическая сумма операндов) с одной и той же оценкой должен иметь ту же оценку;
- при двузначной логике операции многозначной логики совпадают с операциями булевой алгебры.

Результат двух последовательных воздействий факторов  $i$ -го на  $j$ -й с оценкой  $a_{ij}$  и  $j$ -го на  $k$ -й с оценкой  $a_{jk}$  в цепочке 2-х взаимодействий,  $a_{ij} \otimes a_{jk}$  (аналог конъюнкции, или логического произведения в булевой алгебре) определяется по той из двух исходных оценок, которая минимальна по абсолютной величине. Если оба последовательных воздействия имеют оценки разных знаков, то их общий результат будет отрицательным. В противном случае результат положительный.

Результат независимого примитивного воздействия последовательных операций представляется формулой логического умножения

$$a_{ik} = a_{ij} \otimes a_{jk} = \min(|a_{ij}|, |a_{jk}|) \text{sign}(a_{ij} a_{jk}). \quad (1)$$

Результат совместного взаимно дополнительного воздействия группы факторов  $g$  на фактор  $i$ , когда для результата требуется воздействие всех факторов группы, представляется формулой

$$a_{gi} = \min_j \{|a_{ij}|\} \text{sign}\left(\prod_j a_{ij}\right), j \in g. \quad (2)$$

Если на данный фактор  $k$  действуют несколько других факторов, то их совместный эффект  $a_{ik} \oplus a_{jk}$  (аналог дизъюнкции, или логической суммы) может определяться различными способами, в зависимости от того, являются эти факторы взаимодополняющими или взаимно компенсирующими. Параллельное применение операций, воздействующих на какой-либо фактор, может оцениваться по правилам свертки, например, по правилам, применяемым в методе комплексного оценивания [2]. В случае, когда факторы компенсируют друг друга, результат определяется по принципу *доминирования*: результат равен максимальной по абсолютной величине оценке факторов-аргументов

$$a_{ik} \oplus a_{jk} = \min(a_{\max}, \max(|a_{ik}|, |a_{jk}|)) \text{sign}(a_{ik} + a_{jk}). \quad (3)$$

В случае, когда факторы, действующие параллельно, являются дополнительными, результат определяется минимальной по абсолютной величине оценке факторов-аргументов

$$a_{ik} \oplus a_{jk} = \min(a_{\max}, \min(|a_{ik}|, |a_{jk}|)) \text{sign}(a_{ik} + a_{jk}). \quad (4)$$

Для расчета оценки совместного (параллельного) действия факторов, формулирующих оценку  $a_{ij}$  воздействия фактора  $i$  на фактор  $j$ , используется операция группового логического суммирования

$$a_{ij} = \bigoplus_k a_{ik} \otimes a_{kj} \quad (5)$$

При выполнении алгоритмов с рассмотренными наборами операций над оценками взаимодействия исходный набор оценок может распространиться на взаимодействия всех пар факторов. Кроме того, может произойти замена части исходных показателей на значения, большие по абсолютной величине.

Применение операций (1 - 5) образует один шаг рефлексии и отражает изменение состояния в модельном времени. Нетрудно видеть, что при достаточном числе итераций может реализоваться либо сходимость вектора состояния к равновесному значению, либо осциллирующий процесс. Итеративный процесс, использующий преобразования рефлексии, может либо сходиться к вектору оценки состояния с учетом полного набора воздействий, либо породить циклическую последовательность. В последнем случае можно оценить границы изменения оценок факторов.

Варьируя исходные оценки, можно определить степень их участия в формировании полной картины влияний факторов. Кроме того, анализируя результат алгоритма, можно определить противоречивость исходных оценок. Исходная оценка является противоречивой (избыточной), если в результате рефлексий она замещается другой оценкой.

Верификация модели (выбор или уточнение значений коэффициентов матрицы, проводимое с участием эксперта) проводится исходя из требования, чтобы эффект воздействия факторов денежно-финансовой политики на контролируемые факторы в целом соответствовал зависимостям, основанным на статистических данных, а также ожиданиям эксперта. Например, чтобы управляющие воздействия давали положительный результат – приводили к повышению показателей контролируемых факторов (соответствующие коэффициенты влияния имели нужные значение или знак).

Для верификации модели могут использоваться следующие процедуры:

- анализ чувствительности оценки конкретной системной связи к задаваемым вариациям оценок примитивных связей;

- поиск связей, зависящих от конкретной примитивной связи для заданного числа рефлексий;

- поиск примитивных связей, влияющих на данную системную связь за заданное число рефлексий.

Верификацию можно рассматривать как решение обратной задачи, когда по оценкам наблюдаемых зависимостей восстанавливаются оценки примитивных зависимостей, которые затем могут использоваться для расчета изменения равновесных оценок состояния системы в ответ на управляющие воздействия.

## 1. Сценарии верификации

Все перечисленные процедуры верификации модели было бы удобно проводить, имея на руках некий программный инструмент, обладающий способностью запоминать и отслеживать указанные вариации примитивных связей. Причем было бы лучше, если отслеживать вариации можно было визуально. В идеале исследователю необходимо добиться не только количественного, но и качественного видения всей или максимально исчерпывающей картины зависимости «системных» связей факторов от «примитивных» связей. Верификацию модели можно рассматривать как совокупность некоторого (возможно – довольно большого) множества вычислительных экспериментов, для которых организован *эргономичный способ навигации* между различными вариантами их входных данных (и, соответственно, выходных результатов). Важно отметить, что, говоря о входных данных, мы имеем в виду не только количественные меры примитивных связей, но и их *структуру*. Изменение структуры связей означает удаление одних связей и добавление других.

Разберем по порядку перечисленные выше процедуры верификации и постараемся рассмотреть, что означает в каждом случае совокупность вычислительных экспериментов, как она описывается, в каком виде и в каком смысле возможна указанная навигация между экземплярами этой совокупности.

1. Анализ чувствительности оценки конкретной системной связи к задаваемым вариациям оценок примитивных связей.

В данном случае речь идет об обычных функциональных зависимостях: изменяем одну или несколько числовых величин – количественные оценки «примитивных» связей, и наблюдаем, как при этом изменяются вычисленные оценки конкретной «системной» связи.

Часто речь может идти не о том, чтобы провести исчерпывающие вычисления и построить подробную картину функциональной зависимости. Такие вычисления имеют смысл в высокоточном физическом эксперименте, мы же постоянно говорим лишь об оценках. В таких случаях имеет смысл задать лишь некоторое – как правило, небольшое число вариаций численного значения для некоторых «примитивных» связей, и посмотреть на изменение величины интересующей «системной» связи.

В этом случае пользователю желательно иметь не график функции, а удобный интерфейс, позволяющий просто задать вариации одних величин и увидеть отклик другой величины. При этом интересно было бы «поиграть» с разными комбинациями этих вариаций. Однако укажем, что полный – и, следовательно, автоматический – перебор таких комбинаций не имеет большого смысла. Скорее имеет смысл говорить об *эргономичном инструменте*, предоставленном исследователю для разностороннего экспериментирования с моделью.

2. Поиск связей, зависящих от конкретной примитивной связи для заданного числа рефлексий.

В этом случае верифицирующий вычислительный эксперимент состоит в том, что исследователь варьирует только одну примитивную связь и отслеживает изменение *всех* вычисленных (для заданного числа рефлексий) оценочных значений для «системных» связей.

3. Поиск примитивных связей, влияющих на данную системную связь за заданное число рефлексий.

Нам представляется, что это наиболее интересный случай. Дело в том, что в предыдущих двух сценариях верификационных вычислительных экспериментов мы не меняли саму структуру примитивных связей, варьируя тем

или иным способом лишь их численные величины (оценки). В данном же случае исследователю требуется изменять саму структуру графа связей: удалять одни связи и добавлять новые.

В этом случае уже вообще сложно говорить о функциональных зависимостях, во всяком случае – в обычном понимании. Математическая функция есть зависимость одних числовых величин от других. В нашей ситуации в качестве «аргумента» выступает не число и не совокупность чисел, а *структура* – граф связей. В данном виде вычислительного эксперимента нас интересует, как изменяется выбранная «системная» связь при изменении структуры графа «примитивных» связей.

Совершенно ясно, что в данной ситуации вообще бессмысленно говорить о каком-то автоматическом и прямом переборе вариантов. Даже если это несложно организовать алгоритмически (в программе), возникает вопрос – как представить результаты проведенных при этом переборе вычислений наглядным и обозримым образом.

Получается, что в такой ситуации снова уместнее поставить задачу предоставить исследователю некоторый программный инструмент с эргономичным интерфейсом, который позволит последовательно ставить эксперименты с некоторыми *единичными вариациями* графа примитивных связей, наблюдать получаемые результаты, запоминать для каждого эксперимента как структуру графа (конфигурацию связей), так и полученный результат (назовем это *протоколом* эксперимента), а затем иметь возможность свободной навигации между протоколами проведенных экспериментов.

## 2. Графический редактор как программный инструмент верификации

Таким образом, мы приходим к следующей постановке задачи. Для эффективного проведения верификации моделей рассматриваемого типа с учетом различных возможных сценариев, требуется программное обеспечение с эргономичным пользовательским интерфейсом, в основе которого лежит система визуального создания и отображения структур типа математических графов. Далее будем называть эту систему *графическим редактором*, хотя ее

функции, как мы увидим, идут значительно дальше, чем создание визуальных конструкций.

Сразу сделаем следующее предположение: не имеет большого смысла отображать целиком весь граф примитивных связей, это не делает работу исследователя удобней и проще. Мы пришли к выводу, что полный граф связей следует разбить на коллекцию небольших, так называемых «локальных» графов, и пользователь должен иметь возможность свободно манипулировать этими локальными графами и переходить от одного такого графа из коллекции к другому.

Мы различаем два вида локальных графов: графы, описывающие *все* имеющиеся зависимости некоторого одного (текущего) фактора от других факторов, и графы, описывающие *все* имеющиеся зависимости других факторов от данного фактора. Назовем графы первого типа графами *входящего* типа – так как *все* ребра в этих графах входят в узел текущего фактора. Аналогично, графы второго типа уместно назвать графами *исходящего* типа – так как все ребра в этих графах исходят из узла текущего фактора. Можно предложить и другие термины: граф прямых влияний для и граф обратных влияний; это дело вкуса.

При этом не забываем о том, что зависимости в моделях рассматриваемого типа могут быть взаимно дополняющими и взаимно обусловленными. Фактически, взаимная дополнительность означает независимость воздействия нескольких факторов на данный фактор, в то время как взаимная обусловленность означает, что несколько факторов неявно объединены в некоторую *группу*, и именно это группа, со своим оценочным значением влияния, оказывает воздействие на рассматриваемый фактор.

Действие взаимно дополнительных факторов есть дизъюнкция или логическое сложение – в смысле, определенном в главе 1 в формуле (1); действие взаимно обусловленных факторов есть конъюнкция или логическое умножение – в смысле, определенном в главе 1 в формуле (2).

Рассмотрим следующую совокупность факторов, выбранных для когнитивного графа модели социально – экономического развития страны [3]:

1. государственные антикризисные меры;

2. внешнеторговые ограничения;
3. регулирование цен и тарифов;
4. смягчение кредитно-денежной политики;
5. ключевая ставка;
6. кредитные и депозитные ставки;
7. розничное кредитование;
8. инфляционные риски;
9. укрепление рубля;
- 10.денежная масса M2;
- 11.потребительские цены;
- 12.потребительский спрос;
- 13.потребительские расходы;
- 14.заработная плата;
- 15.сбережения в рублях;
- 16.кредитование производства;
- 17.новые заказы;
- 18.экономическая активность;
- 19.загрузка производственных мощностей;
- 20.инвестиционный спрос;
- 21.загрузка рабочей силы;
- 22.неполная занятость;
- 23.выпуск, ВВП;
- 24.реальные доходы;
- 25.численность населения;
- 26.прожиточный минимум;
- 27.уровень бедности;
- 28.дефицит бюджета;
- 29.спрос на импорт;
- 30.экспорт;
- 31.замещение внешних источников финансирования внутренними;
- 32.финансовый рынок;
- 33.рублевая долговая нагрузка;
- 34.цены на энергоносители;
- 35.внешние шоки;
- 36.внешнеэкономическая конъюнктура.

Из них часть факторов, перечисленных ниже, принята в качестве контрольной:

- 22.неполная занятость;
- 23.выпуск, ВВП;
- 24.реальные доходы;
- 25.численность населения;
- 26.уровень бедности;
- 27.дефицит бюджета;
- 28.спрос на импорт;
- 29.экспорт;

Теперь приведем *фрагмент* системы взаимной зависимости факторов (численный коэффициент означает, что связь ограничена по мо-

дулю сверху соответствующей пропускной способностью канала влияния):

1. *государственные антикризисные меры* = 5 (инфляционные риски)  $\vee$  7# (внешние шоки)  $\wedge$  (-5) (внешнеэкономическая конъюнктура)
2. *внешнеторговые ограничения* = 4 (государственные антикризисные меры)  $\wedge$  -5 (внешнеэкономическая конъюнктура)
3. *регулирование цен и тарифов* = 8 (государственные антикризисные меры)  $\wedge$  -2 (смягчение кредитно-денежной политики)  $\vee$  5 (инфляционные риски)
4. *смягчение кредитно-денежной политики* = -3 (инфляционные риски)  $\vee$  -2 (ослабление рубля)  $\vee$  5 (внешнеэкономическая конъюнктура)
5. *ключевая ставка* = (-4) (смягчение кредитно-денежной политики)
6. *кредитные и депозитные ставки* = 8 (ключевая ставка)  $\vee$  3 (инфляционные риски)
7. *розничное кредитование* = -5 (кредитные и депозитные ставки)  $\vee$  4 (потребительские расходы)
8. *инфляционные риски* = (-3) (регулирование цен и тарифов)  $\vee$  4 (ослабление рубля)  $\vee$  5 (денежная масса M2)  $\vee$  3 (рублевая долговая нагрузка)  $\vee$  7 (внешние шоки)
9. *укрепление рубля* = (-4) (инфляционные риски)  $\vee$  (-5) (денежная масса M2)  $\vee$  4 (цены на энергоносители)
10. *денежная масса M2* = 4 (ослабление рубля)  $\vee$  3 (реальные доходы)  $\vee$  6 (экспорт)
- ...
34. *цены на энергоносители* = 5 (инфляционные риски)  $\vee$  5 (финансовый рынок)

Для зависимости 1 мы имеем, что фактор «государственные антикризисные меры» (назовем его для краткости аббревиатурой «ГАМ») зависит от взаимно компенсирующих (то есть – логическим сложением в «нашем» смысле) фактора «инфляционные риски» («ИР») и группы взаимно обусловленных (дополнительных) факторов «внешние шоки» («ВШ») и «внешнеэкономическая конъюнктура» («ВЭК»). Назовем эту группу «Г1». Взаимная обусловленность последних факторов задается опера-

тором конъюнкции и означает: внешние шоки не имеют значения без внешнеэкономической конъюнктуры и взаимно обратно.

Графически данную зависимость можно изобразить как совокупность двух очень простых локальных графов (Рис. 1). Мы предполагаем такую функциональность нашего графического редактора (можно также говорить о визуальном интерфейсе пользователя): при клике на квадратик «ИР» тем или иным способом (на этом же графическом «полотне» либо в отдельном окне) открывается локальный граф (или несколько локальных графов, с учетом наличия объединения факторов в группы при взаимно обусловленном воздействии) для фактора «инфляционные риски».

Теперь надо не забыть о втором типе локальных графов: графах исходящего типа или графах обратных влияний. Так, если мы говорим о факторе «инфляционные риски» («ИР»), то граф исходящего типа должен показать, что данный фактор влияет на факторы «укрепление рубля» («УР») и «цены на энергоносители» («ЦнЭ»). Тогда получим граф, как на Рис. 2.

Если рассмотреть фактор «государственные антикризисные меры» («ГАМ»), то увидим, что в зависимости (2) он входит в группу с фактором «внешнеэкономическая конъюнктура» («ВЭК») и

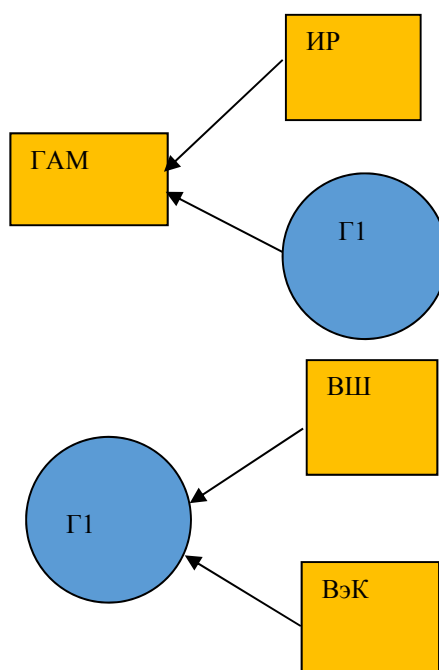


Рис. 1. Воздействие на фактор «государственные антикризисные меры»

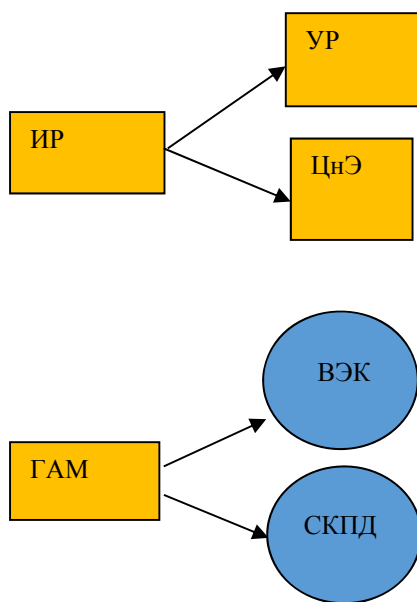


Рис. 2. Воздействие факторов "инфляционные риски" и "государственные антикризисные меры"

в зависимости (3) входит в группу с фактором «*смягчение кредитно-денежной политики*» («СКПД»). Локальный граф, описывающий эти вхождения, показан на Рис. 2 внизу.

Важно: в общем случае следует ожидать, что графы обратных влияний (исходящего типа), так же, как и графы прямых влияний (входящего типа), могут включать как отдельные факторы, так и группы.

Кроме того, следует понимать, что возможно существование групп, которые состоят из одних и тех же факторов, но это разные группы, поскольку оказывают взаимно компенсирующее влияние на разные факторы. В таких случаях желательно отслеживать, чтобы эти группы получили отличающиеся названия (во всяком случае, разные идентификаторы внутри программы).

Локальные графы составляют основу функциональности графического редактора, но требуется рассмотреть, как все работает целиком, поскольку заявленная задача редактора – верификация моделей многофакторного взаимодействия.

### 3. Полный цикл работы пользователя графического редактора

Полный цикл (сценарий) работы исследователя, занятого верификацией моделей рассматриваемого типа, предварительно, в главных чертах, должен состоять из следующих этапов.

1. Создание и регистрация всех имеющихся факторов. Факторы составляют узлы полного графа влияний (мы договорились, что не обязательно его отображать целиком, но мы всегда помним о его существовании). Просмотр и редактирование имеющихся факторов важен для исследователя независимо от связей влияния между этими факторами. Для этого должен быть предусмотрен свой графический интерфейс.

2. Создание и регистрация всех формул взаимного влияния факторов. Это предполагает отрисовку локальных графов, рассмотренных выше. Каждому элементарному взаимодействию между факторами (либо фактором и группой взаимно дополнительных факторов) соответствует ребро локального графа – попросту, «стрелка». Графический интерфейс должен предусматривать возможность редактирования свойств «стрелки». Например, при клике манипулятором «мышь» по графическому примитиву «стрелки» должно открываться окошко редактирования математической или логической формулы, описывающей данное взаимодействие.

3. Проведение некоей автоматизированной (программной) проверки целостности модели. Это означает примерно следующее: не должно оставаться «подвешенных» факторов (либо пользовательский интерфейс должен как-то выделять эти факторы как незадействованные в модели), должны быть заданы все формулы взаимодействия.

4. Проведение серий вычислительных экспериментов, в частности, верификации модели по одному из сценариев, описанных выше в главе 2. Не вдаваясь в детали каждого из этих сценариев, можно выделить то общее, что их объединяет. Во всех сценариях мы производим изменения в одном или нескольких локальных графах влияния факторов (как правило, в небольшом числе). Некоторые связи можем просто «отключать» либо добавлять новые. Модифицировав тем или иным образом некоторую часть связей влияния факторов, исследователь задает число рефлексий и «запускает» расчет оценочных значений для «системных» связей между факторами.

В главе 1 было отмечено, что в этой ситуации должен существовать *протокол* проводимой последовательности вычислительных экс-

периментов. Протокол представляет собой структуру данных в памяти компьютера. Используя формат разметки данных JSON [3], эту структуру можно преобразовать к удобному текстовому виду. В принципе, опытный пользователь системы может работать напрямую с этим текстовым представлением. Однако желательно, чтобы протокол имел визуальное представление.

В визуальном плане протокол *серии* вычислительных экспериментов можно представить как граф связанных *конфигураций* эксперимента. Конфигурация эксперимента – это и есть полный граф влияния факторов, точнее, его формализованное описание. Строго говоря, внося даже единичное изменение на уровне отдельной связи, мы каждый раз создаем новую конфигурацию. Этой конфигурации можно присвоить некоторое название или идентификатор. Внося изменения шаг за шагом, мы создаем некоторую сеть (граф) *именованных* конфигураций.

Смысл визуального представления протокола вычислительного эксперимента (или серии экспериментов) состоит в первую очередь в том, чтобы иметь возможность оперативно переходить от одной конфигурации к другой – осуществлять *навигацию* между конфигурациями. Организация такой навигации, конечно, требует тщательной проработки прежде всего на концептуальном уровне, не говоря о значительном ожидаемом объеме программирования.

#### 4. Нарботки в области «графического редактора»

Сформулированные в предыдущих главах постановки задач могут выглядеть довольно амбициозными, и закономерно возникает вопрос о посильности реализации этих задач в рамках академического процесса (а не высокобюджетного ИТ – бизнеса).

Рискнем утверждать, что у нас уже имеется неплохой задел. В рамках серии государственных контрактных работ один из авторов стал разработчиком программно-вычислительного комплекса, одним из компонентов которого как раз и стал графический редактор, принципиально похожий на вышеописанную систему [4-6].

Отличительная черта разработанного графического редактора (возможно, пока более

уместно говорить о его макете, потому что функциональность ограничена) – он реализован в среде веб-браузера и написан на языке браузерного программирования JavaScript. Это сразу дает много плюсов как с точки зрения пользователя – фактически не нужно устанавливать отдельное специализированное программное обеспечение, достаточно открыть в браузере подготовленную веб-страницу, так и с точки зрения разработки – потому что в настоящее время уже существует огромное количество и появляется все больше новых *специализированных библиотек*, написанных на JavaScript. В частности, существуют довольно мощные графические библиотеки (мы использовали Raphael [7]). Существуют и библиотеки численных методов [8].

В указанных работах мы не использовали математические библиотеки, так как роль вычислительного блока выполнял программный комплекс Octave [9, 10], который получал «задание» на вычисления от интерфейсной веб-страницы через веб-сервер. Но теперь мы готовы рассмотреть математические возможности самого JavaScript, в этом случае пользовательское приложение будет существовать, как мы заявили чуть выше, вообще без отдельной программы – просто набор файлов, которые пользователь скопирует себе на компьютер, и будет загружать страницу с графическим редактором прямо в своем браузере.

Необходимо добавить, что существующий макет графического редактора с самого начала был нацелен как раз на отрисовку графов.

#### Заключение

Рефлексивные модели сложного взаимодействия факторов в экономике проявили себя как математические конструкции во многом качественно нового типа. Эти модели, имея изначально вычислительный, компьютерный характер, оказались во многом отличными, например, от задач численного интегрирования дифференциальных уравнений. Значительная доля прикладных математических моделей в естественных науках и технике, действительно, есть эволюционные модели, описываемые системами дифференциальных уравнений. При этом потребителя такой модели интересует, в конечном счете, про-



сто траектория системы, максимум – пучок траекторий. Получение пучка траекторий может оказаться вычислительно довольно трудоемкой операцией, но идеологически она проста: есть начальные (либо граничные) условия, представляющие собой некоторую область в N-мерном пространстве эволюционных переменных, и мы должны с тем или иным дискретным шагом перебрать точки в этой области.

В случае моделей рефлексивного типа на графах пространство перебора значительно расширяется, мало того – оно качественно изменяется. Например, добавлять или удалять те или иные связи влияния – все равно, что добавлять или удалять члены в правой части дифференциальных уравнений. Получается, что мы изменяем не только начальные условия, но и варьируем саму модель. При этом возможность некоего «подробного» перебора вариантов (аналогично получению пучку траекторий) вообще теряет смысл.

В этой ситуации становится уместным говорить о вычислительных моделях нового типа – это *человеко-машинное* моделирование, или, иначе, моделирование при помощи человеко-машинных систем. Предлагаемый «графический редактор» и есть прототип такой системы. Описанные выше сценарии верификации рефлексивных моделей на графах отчасти опираются на некий «прямой перебор», отчасти – на интуицию и плохо формализуемую логику исследователя, проводящего вычислительный эксперимент (точнее, серию экспериментов) с применением варьирования связей влияния.

Можем добавить, что в беседе авторов с профессором Г. Г. Малинецким для такого процесса математического моделирования родился термин «*дискретное вариационное исчисление*». В самом деле, проводя верификацию моделей по описанным сценариям, мы как бы наощупь изучаем совместное пространство начальных условий и

самих моделей взаимодействия факторов в некоторой окрестности начальной «траектории» системы (в данном случае – экономической). Для практической реализации «дискретного вариационного исчисления», как представляется, как раз и нужны человеко-машинные системы моделирования, близкие по духу к описанному «графическому редактору».

## Литература

1. Гусев В.Б., Исаева Н.А. Рефлексивные процедуры анализа экспертных данных // Информационные технологии и вычислительные системы. 2016. № 2. С. 31-35.
2. Гусев В.Б., Павельев В.В. Использование непрерывных шкал при оценивании и принятии решений в сложных проблемных ситуациях: научное издание / В.Б. Гусев, В.В. Павельев. – М.: ИПУ РАН, 2013. – 118 с.
3. Гусев В.Б., Исаева Н.А. Анализ моделей управления на основе экспертных данных. Москва: ИПУ РАН, 2017. – 116 с.
4. Официальная страница формата JSON на русском языке, URL: <http://json.org/json-ru.html>.
5. Зухба Р.Д., Куракин П.В., Малинецкий Г. Г., Махов С. А., Митин Н. А., Торопыгина С.А. Система моделирования «КОСКОН» как инструмент поддержки принятия решений в космической отрасли. Препринты ИПМ им. М. В. Келдыша. — 2015. — № 113. — 36 с. — URL: <http://library.keldysh.ru/preprint.asp?id=2015-113>.
6. П. В. Куракин. «Новая программная архитектура для специализированных систем математических расчетов». Информационные технологии и вычислительные системы, 2016 г, № 2 стр. 66 – 74.
7. Куракин П. В. «Специализированные системы математических расчетов нового поколения». Программные системы и вычислительные методы – 2016 г, №1(14), стр. 80 – 94. DOI: 10.7256/2305-6061.2016.1.17997
8. Официальный веб-сайт проекта Raphael, URL: <http://raphaeljs.com/>
9. John Mueller, Four Serious Math Libraries for JavaScript (<https://blog.smartbear.com/testing/four-serious-math-libraries-for-javascript/>).
10. Octave (статья в Wikipedia), URL: [https://ru.wikipedia.org/wiki/GNU\\_Octave](https://ru.wikipedia.org/wiki/GNU_Octave).
11. Octave (официальная страница проекта) , URL: <https://www.gnu.org/software/octave/>

**Гусев Владислав Борисович.** Институт проблем управления им. В. А. Трапезникова Российской академии наук, г. Москва, Россия. Ведущий научный сотрудник, кандидат физико-математических наук. Количество печатных работ: 176 (в т.ч. 10 монографий). Область научных интересов: теория систем, математическая экономика, вычислительная математика, информационные технологии.

**Куракин Павел Вячеславович.** Институт проблем управления им. В. А. Трапезникова Российской академии наук, г. Москва, Россия. Научный сотрудник. Количество печатных работ: 50. Область научных интересов: явления социальной самоорганизации, математическая экономика, автоматизация вычислительного эксперимента. E-mail: [pvkurakin@yandex.ru](mailto:pvkurakin@yandex.ru)

## Software instruments for verification of models of complex interactions of economic factors

V. B. Gusev, P. V. Kurakin

V. A. Trapeznikov Institute of Control Sciences of Russian Academy of Sciences, Moscow, Russia

**Abstract.** The paper suggests an original approach for development man – computer technology of verification of models of complex interaction of economic factors. We argue that the task of such a verification demands qualitatively new approaches and concepts compared to traditional problems of numerical modeling. We describe partially implemented software solution which pretends to make a response to existing challenge. The solution is based on visual design of exemplars of appropriate problems in the medium of specialized visual editor.

**Keywords:** economy, interaction of factors, mutually complementary factors, mutually compensating factors, visual editor, computational experiment.

**DOI** 10.14357/20718632190106

### References

1. Gusev V. B., Isaeva N. A. Refleksivnie protsedury analiza ekspertnykh dannykh [Reflexive procedures of analysis of experts' data]. Informatsionnie tehnologii i vychislitelnie sistemy [Information technologies and computational systems]. 2016. 2: 31-35.
2. Gusev V. B., Pavelyev V. V. Ispolzovanie neprerivnykh shkal pri otsenivanii i prinyatii reshenii v slozhnykh problemnykh situatsiyah: nauchnoe izdanie [Using continuous scales for estimating and solution making in complex problem situations: scientific issue]. Moskva, IPU RAN [Moscow, Institute of control sciences, Russian Academy of Sciences]. 2013, 118 pp.
3. Gusev V. B., Isaeva N. A. Analiz modelei upravleniya na osnove ekspertnykh dannykh [Analysis of control models based on experts' data Moskva], IPU RAN [Moscow, Institute of control sciences, Russian Academy of Sciences]. 2017, 116 pp.
4. Ofitsialnaya stranitsa formata JSON na russkom yazyke [Official page of JSON project in Russian]. Available at: <http://json.org/json-ru.html>.
5. Zuhba R. D., Kurakin P. V., Malinetskii G. G., Mahov S. A., Mitin N. A., Toropygina S. A. Systema modelirovaniya "KOSKON" kak instrument podderzhki prinyatiya resheniy v kosmicheskoi otrasli [The simulation system "COSCON" as a tool for decision support in the space industry]. Preprinty IPM im. M. V. Keldysha [Preprints by Keldysh Institute of Applied Mathematics]. 2015: 113. 36 pp. Available at: <http://library.keldysh.ru/preprint.asp?id=2015-113>.
6. P. V. Kurakin. Novaya programmnaya arhitektura dlya spetsializirovannykh system matematicheskikh raschetov [A new software architecture for specialized systems of mathematical computations]. Informatsionnie tehnologii i vychislitelnie sistemy [Information technologies and computation systems]. 2016: 2, 66 – 74.
7. P. V. Kurakin. Spetsializirovannye sistemi matematicheskikh raschetov novogo pkoleniya [New generation of specialized systems for mathematical computation]. Programmnye sistemi i vychislitelnye metody [Software systems and computational methods]. 2016:1, 80 – 94. DOI: 10.7256/2305-6061.2016.1.17997.
8. Ofitsialniy web – sait proekta Raphael [Official web – site of Raphael project]. Available at: <http://raphaeljs.com/>.
9. John Mueller, Four Serious Math Libraries for JavaScript. Available at: <https://blog.smartbear.com/testing/four-serious-math-libraries-for-javascript/>.
10. Octave (article at Wikipedia), Available at: [https://ru.wikipedia.org/wiki/GNU\\_Octave](https://ru.wikipedia.org/wiki/GNU_Octave).
11. Octave (official web – page of the project), Available at: <https://www.gnu.org/software/octave/>

**Gusev V. B.** V. A. Trapeznikov Institute of Control Sciences, Russian Academy of Sciences, Moscow, Russia. Leading research worker, Ph. D. 176 published scientific papers (including 10 monographs). Research interests: theory of systems, mathematical economy, computational mathematics, information technologies.

**Kurakin P. V.** V. A. Trapeznikov Institute of Control Sciences, Russian Academy of Sciences, Moscow, Russia. Research worker. 50 published scientific papers. Research interests: social self-organization, economy, automatization of computational experiment. Email: [pvkurakin@yandex.ru](mailto:pvkurakin@yandex.ru)