

Приложение на основе гибридной имитационной модели локальной предпринимательской сети

Application based the hybrid simulation model of a local entrepreneurial network

doi 10.26310/2071-3010.2021.274.8.005



А. В. Апанасенко,

аспирант, кафедра анализа систем и принятия решений, ИнЭУ, Уральский федеральный университет им. первого Президента России Б. Н. Ельцина
✉ stacy-chan@yandex.ru

A. V. Apanasenko,

postgraduate student, department of systems analysis and decision-making, GSEM, Ural federal university named after the first President of Russia B. N. Yeltsin

В этой статье представлено автономное Java-приложение, разработанное на основе гибридной имитационной модели предпринимательской сети локального сообщества. Гибридная модель сочетает в себе системную динамику и агентно-ориентированные методы. Это позволяет рассматривать систему под разными углами. Уровень системной динамики показывает денежные потоки внутри каждого предприятия, участвующего в сети. На агентном уровне представлена система в целом, что позволяет сформулировать и оценить стратегию ее развития, а также учесть синергетический эффект. Приложение на основе модели создано с помощью Anylogic. Оно содержит три экрана: настройка управляющих параметров, агентный уровень и системно-динамический уровень. В качестве входных данных используется матрица платежей локального сообщества. Управляющие параметры позволяют выбирать методы расширения сети, использования излишков и включение дополнительных средств расчета. Различные комбинации управляющих параметров дают возможность определить лучшую стратегию, используя имитационные эксперименты. Оценка основана на сравнении итоговых показателей и визуализации. Описанное приложение можно использовать для поддержки принятия решений при муниципальном управлении.

This article describes a standalone java application developed based on the hybrid simulation model of a local community entrepreneurial network. The hybrid model combines system-dynamic and agent-based approaches. This allows looking at the system from different angles. System dynamics level shows cash flows within each enterprise participating in the network. The agent level shows the system as a whole, allows to draw up and evaluate a strategy for its development, take the synergistic effect into account. Model-based application is built with Anylogic. It contains three screens: setting control parameters, agent level, system-dynamic level. The matrix of local community cash transactions is used as input data. The control parameters allow choosing methods of expanding the network, using surpluses and including additional means of payment. Various control parameters combinations allow making a simulation experiment for choosing a better strategy. The assessment is based on comparison of final indicators and visualizations. The described application can be used to support decision-making for municipality management.

Ключевые слова: гибридная модель, системная динамика, агентное моделирование, java-приложение, предпринимательские сети, локальное сообщество, дополнительные средства расчетов.

Keywords: hybrid model, simulation modeling, system dynamics, agent-based modeling, java application, entrepreneurial networks, local community, additional means of payment.

Введение

Существуют различные методы имитационного моделирования. Каждый из них имеет свои особенности, что обуславливает различные области их применения. Так дискретно-событийное моделирование применяется для имитации производственных процессов и отражает «узкие места» производства. Системная динамика применяется для моделирования финансовых, производственных и прочих ресурсных потоков, иллюстрирует взаимосвязи и взаимовлияние элементов системы. Агентное моделирование имитирует деятельность системы как совокупности индивидуально действующих агентов, воспроизводя их поведение по заданным алгоритмам. Это позволяет увидеть как жизненный цикл отдельных элементов системы, так и синергетический эффект от их сосуществования.

Но когда речь идет о моделировании сложной социально-экономической системы, возникает потребность применить сразу несколько методов, чтобы воспользоваться их достижениями. Одним из способов реализации этого подхода является создание гибридных моделей. «Гибридными называются модели, использующие различные подходы, стили, парадигмы моделирования в рамках одной разработки

для построения наиболее адекватной и результативной модели» [1].

Изучению гибридных имитационных моделей посвящены работы как иностранных, так и отечественных ученых. Бреисфорд, Чучилов и Лиев сформулировали необходимость такого подхода [2]. Джахангириан отметил его перспективность для моделирования сложных корпоративных систем [3]. Мустафи и Пауэлл составили широкую таксономию для их классификации [4].

С точки зрения практического применения, наиболее популярным типом гибридных моделей являются сочетающие дискретно-событийное и системно-динамическое моделирование [5]. Но в последнее время приобретают популярность и другие сочетания методов с различной степенью интегрированности. Примерами таких работ являются публикации Хэджина [6] и Кикхефера [7] с коллегами, где описана интеграция моделей системной динамики с агентными. А также Уоллентина и Ньюирса, где реализована возможность переключения между моделями этих же типов [8].

Среди отечественных ученых можно выделить Кателевского [9] и Аكوпова [10]. В работах этих авторов отражены два противоположных способа ком-

бинирования методов моделирования. Каталевкий использует агентное моделирование на микроуровне, отражая индивидуальное поведение экономических агентов, а системную динамику — на макроуровне для описания окружающей их среды. Акопов же наоборот, использует системную динамику для моделирования процессов внутри предприятий-агентов, а агентное моделирование для описания их взаимодействия.

В целом анализ источников показал, что даже при наличии примеров реализации гибридных моделей, этот подход представлен значительно меньше, чем использование любого конкретного метода моделирования. Это обусловлено как относительной новизной самой идеи, так и сложностью ее технической реализации. Далеко не все системы моделирования позволяют сочетать различные методы встроенными средствами. Исключением является AnyLogic, в который такая возможность была заложена изначально. Эта среда моделирования обеспечивает взаимодействие и синхронизацию между моделями [11]. Таким образом, сейчас гибридное имитационное моделирование является перспективным подходом, особенно при создании моделей сложных систем.

Материалы и методы

Одной из областей применения компьютерного имитационного моделирования вообще и гибридного в частности является использование моделей как инструмента поддержки принятия решений. К примеру, при управлении локальным сообществом возникает необходимость выбора оптимального решения и оценки эффективности принятых управленческих решений. В качестве инструментов поддержки принятия решений используются методы клеточных автоматов [12], агентно ориентированные [13] и системно-динамические [14]. На основе данных, полученных при моделировании, ответственные лица могут снизить неопределенность и принять более взвешенное и обоснованное решение. Конечно, имитационная модель, применяющаяся таким образом должна учитывать

особенности конкретного локального сообщества, характер используемых статистических данных и уже имеющийся опыт местного самоуправления, а также, технические возможности проведения модельных расчетов. Тогда она будет эффективным инструментом, положительно влияющим на развитие локального сообщества.

Гибридное моделирование было применено для создания имитационной модели предпринимательской сети местного сообщества в среде имитационного моделирования Anylogic. Первая версия модели описана в статье [15]. В рамках последующих исследований модель была доработана и на ее основе создано автономное java-приложение, которое и описывается в данной статье. Приложение имеет в своей основе гибридную имитационную модель, один уровень которой является агентным, а второй — системно-динамическим. Такая структура позволяет взять лучшее от двух методов и визуально разделить информационные слои модели. На рис. 1 представлена концептуальная схема модели.

Реализация в виде java-приложения позволяет, в свою очередь, решить проблему зависимости от специализированного программного обеспечения, которым является среда моделирования. В нотации Anylogic алгоритм работы модели можно описать следующим образом:

1. При запуске:
 - a) считываются значения элементов управления и исходные данные моделирования из MS Excel;
 - b) создается популяция из 20 агентов;
 - c) заполняются исходные данные агентов;
 - d) создаются графики по данным существующих агентов.
2. Перед выполнением каждого шага:
 - a) пересоздаются связи между агентами, если их существование обосновано данными;
 - b) генерируются фактические финансовые показатели на основе плановых;
 - c) рассчитываются суммарные расходы агентов и недостаток капитала.

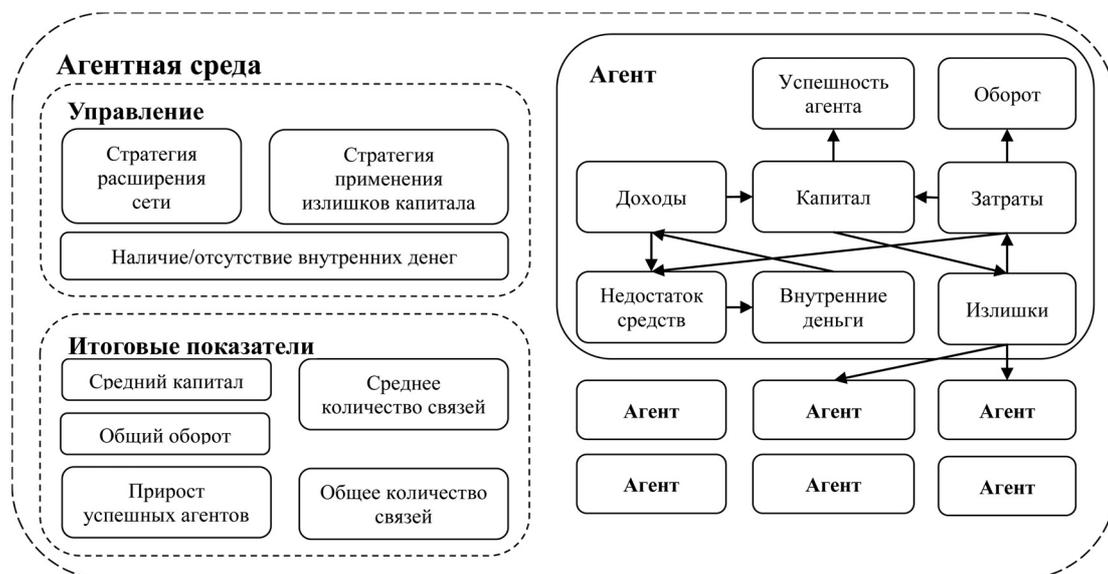


Рис. 1. Концептуальная схема гибридной имитационной модели



Рис. 2. Первый экран приложения — уровень управляющих параметров

3. На каждом шаге моделирования:
 - а) совершается платеж — пересылка средств от одних агентов к другим;
 - б) рассчитываются суммарные доходы агентов;
 - в) пересчитываются прочие данные агентов, включая их состояние;
 - г) пересчитываются итоговые показатели, отражающие состояние всей сети.
4. После каждого шага моделирования добавляется агент к популяции, если сеть может себе это позволить по условиям, строятся графики по его данным.

Основная часть статьи

Реализованное приложение состоит из трех экранов. Первый экран содержит настройки управляющих параметров модели. Второй и третий соответствуют уровням модели — агентному и системно-динамическому. Скриншот первого экрана приведен на рис. 2.

Здесь можно настроить управляющие параметры модели:

- Growth Method — стратегия расширения сети. Может принимать значения: Cheap First — в первую очередь в сеть войдут предприятия с наименьшей стоимостью присоединения, что позволит

максимально быстро увеличивать размеры сети, и BigFirst — в первую очередь будут присоединены предприятия, обладающие наибольшим количеством связей, а значит, самым высоким потенциалом для развития сети.

- Capital Use — стратегия применения излишков капитала. Допустимые значения: to Add — излишки пойдут на присоединение к сети новых агентов, что обеспечит приток новых доходов для участников сети, и to Replenish — излишки будут применены для пополнения капитала убыточных участников сети, что позволит избежать разрыва налаженных цепочек обмена.
- insideMoneyUse — использование внутренних денег, дополнительного средства обмена [16]. Оно позволяет избежать ситуации, когда предприятие, оказавшись убыточным, не имеет средств на совершение обмена и затормаживает деятельность всей сети.

Тут же расположена кнопка для запуска имитационного эксперимента.

Второй экран приложения соответствует агентному уровню модели и является наиболее содержательным (рис. 3). Во-первых, здесь визуализируется состав и структура предпринимательской сети. Предприятия — участники сети обозначены пиктограммами. Их цвет обозначает, насколько предприятие успешно:

- красный — на данном шаге предприятие убыточно, значение его капитала опустилось ниже нуля. Фактически это означает, что участник задолжал своим сетевым партнерам;
- желтый — уровень успеха предприятия — средний. Его капитал выше нуля, но ниже среднего по сети;
- зеленый — предприятие успешно. Уровень его капитала не только имеет положительное значение, но и превышает средний показатель по сети.

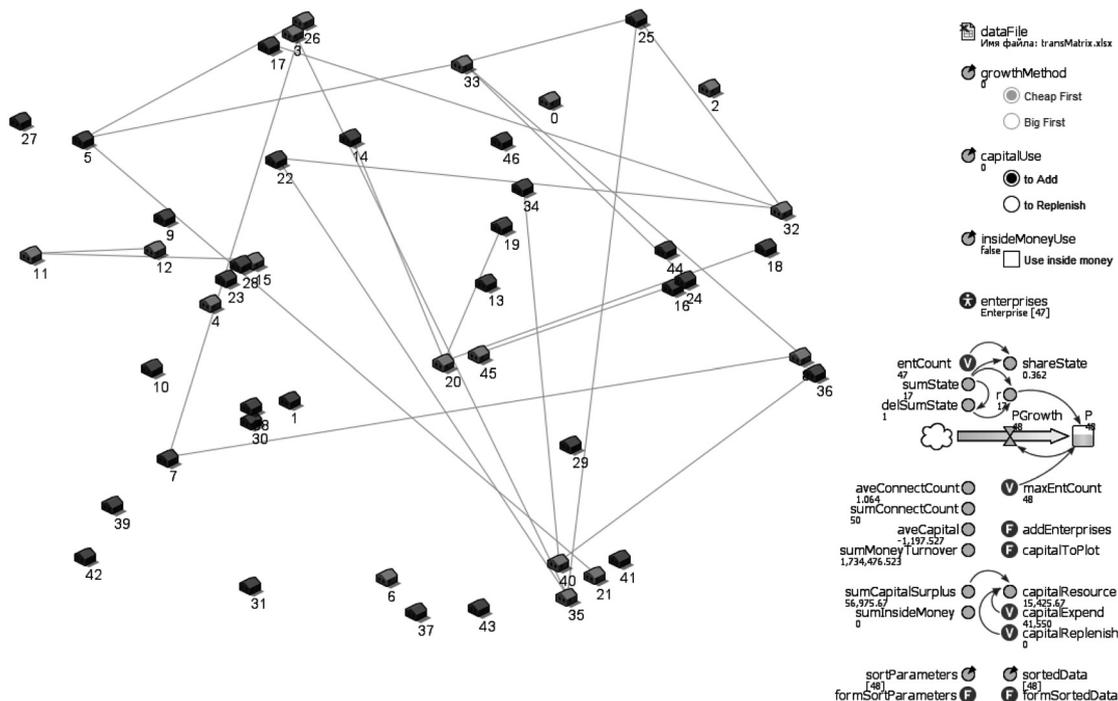


Рис. 3. Второй экран приложения — агентный уровень

Также для каждого участника сети отображается порядковый номер, что позволяет идентифицировать его среди других. Линии, соединяющие пиктограммы участников обозначают связи, т. е. платежи участников на данном шаге. Такая визуализация позволяет проследить в динамике состояние и интенсивность обмена, как каждого участника, так и сети в целом.

Во-вторых, на этом экране расположены элементы, общие для всех агентов системы. Например, отвечающие за работу с исходными данными. Исходными данными для моделирования является матрица плановых взаимоплатежей предприятий муниципалитета за месяц в тысячах денежных единиц. Также матрицу дополняют начальный капитал и максимальное возможное количество связей для каждого предприятия.

Относительно первой версии модели работа с исходными данными в приложении была значительно улучшена. Чтение данных происходит из файла электронной таблицы MS Excel. Изначально переключение между способами расширения предпринимательской сети осуществлялось чтением данных с разных листов электронной таблицы. Такой подход усложнял работу с моделью, так как требовал предварительной подготовки данных. Перед запуском модели исходную таблицу транзакций требовалось отсортировать по выбранному критерию средствами MS Excel. Это противоречит одной из основных идей реализации данного приложения — максимальной автоматизации поддержки принятия решений лицами, управляющими функционированием локальной предпринимательской сети. В новой версии обработка исходных данных полностью автоматизирована. Исходные данные считываются с одного листа и сортируются в зависимости от значения одного из управляющих параметров модели — Growth Method. Сортировка данных для определения порядка добавления участников в сеть происходит программно. Чтение и сортировка данных обеспечиваются dataFile — элементом нотации Anylogic типа «Файл Excel» и двумя функциями Java:

- formSortParameters — считывает столбцы с данными, по которым будет происходить сортировка: начальный капитал и количество связей каждого агента. Результат записывается в параметр sortParameters:

```
maxEntCount=48;
//Читаем столбцы с данными, по которым будет происходить сортировка
double[][] sortParameters=new
double[maxEntCount][3];
for(int i=0; i<maxEntCount; i++){
    for(int j=0; j<3; j++){
        sortParameters[i][j]=dataFile.
getCellNumericValue («basicData», i+1,
j+maxEntCount);
    }
}
//Записываем в первый столбец цифры 1-48,
чтобы после сортировки значений обращаться к
ним как к индексам столбцов
for(int i=0; i<maxEntCount; i++){
    sortParameters[i][0]=i;
}
//Сортировка от меньшего к большему
```

```
if (growthMethod==0){
    java.util.Arrays.sort(sortParameters,
new java.util.Comparator<double[]>(){
    public int compare(double[] a,
double[] b){
        return Double.
compare(a[2], b[2]);
    }
});
}
//Сортировка от большего к меньшему
else if (growthMethod==1){
    java.util.Arrays.sort(sortParameters,
new java.util.Comparator<double[]>(){
    public int compare(double[] a,
double[] b){
        return Double.
compare(b[1], a[1]);
    }
});
}
return sortParameters;
```

- formSortedData — считывает исходную матрицу денежных транзакций и сортирует ее по начальному капиталу или количеству связей каждого агента в зависимости от значения параметра Growth Method. Результат записывается в параметр sortedData:

```
maxEntCount=48;
//Заполняем двумерный массив данными из
Excel
double[][] sortData=new double[maxEntCount]
[maxEntCount];
for(int i=0; i<maxEntCount; i++){
    for(int j=0; j<maxEntCount; j++){
        sortData[i][j]=dataFile.getCell
NumericValue («basicData», i+1,
j+1);
    }
}
//Читаем столбцы с данными, по которым будет происходить сортировка
double[][] sortParameters=formSortParameter
s();
//Записываем строки в массив в правильном
порядке
double[][] stringArray=new
double[maxEntCount][maxEntCount];
for (int i=0; i<maxEntCount; i++){
    stringArray[i]=sortData[(int)
sortParameters[i][0]];
}
//Записываем столбцы в массив в правильном
порядке
double[][] sortedData=new
double[maxEntCount][maxEntCount];
for(int i=0; i<maxEntCount; i++){
    for(int j=0; j<maxEntCount; j++){
        sortedData[i][j]=stringArray[i]
[(int)sortParameters[j][0]];
    }
}
return sortedData;
```

Другими общими элементами являются управляющие параметры модели, интерфейс которых

расположен на первом экране: Growth Method, CapitalUse и insideMoneyUse. Стоит отметить, что значения данных параметров, а значит, условия эксперимента могут быть изменены в процессе моделирования.

Здесь же расположен enterprises — элемент «Популяция агентов» нотации Anylogic, позволяющий реализовать масштабирование и вложенность агентов. Содержит параметры популяции, такие как начальное количество агентов и статистические функции, используемые для расчета итоговых показателей функционирования сети.

Далее следуют итоговые показатели моделирования, перечень которых был расширен по сравнению с первой версией модели. А также, вспомогательные элементы, используемые для их вычисления. Итоговые показатели реализованы через переменные системной динамики. Значения большинства из них рассчитываются с помощью встроенных статистических функции Anylogic:

- aveConnectCount — среднее количество связей агентов;
- sumConnectCount — общее количество связей сети;
- aveCapital — средний капитал;
- sumMoneyTurnover — общий денежный оборот;
- sumCapitalSurplus — общие излишки капитала;
- sumInsideMoney — общее количество использованных внутренних денег;
- sumState — количество агентов в неубыточном состоянии;
- shareState — доля агентов в неубыточном состоянии;
- entCount — текущее количество участников сети;
- r — коэффициент, отражающий прирост жизнеспособности предпринимательской сети, рассчитывается как отношение количества агентов, способных

к обмену, на текущем шаге к этому же показателю на предыдущем.

Перечисленные показатели отражают общее состояние предпринимательской сети как системы. Чем больше среднее и общее количество связей, абсолютное и относительное количество агентов в неубыточном состоянии, а также средний капитал и коэффициент r , тем о более успешной деятельности сети можно говорить. Общий денежный оборот и общее количество использованных внутренних денег отражают интенсивность обмена.

Как видно по алгоритму, описанному выше, одним из основных допущений, на которых строится модель, является возможность присоединения на каждом шаге нового участника сети. Присоединение происходит за счет средств, формируемых из излишков капитала. Для реализации расширения сети используются переменная системной динамики capitalResource и функция Java addEnterprises. Здесь capitalResource — сумма доступная для присоединения новых и поддержки убыточных участников сети, рассчитывается как разность между излишками капитала и уже потраченными средствами:

$$\text{capitalResource}(t) = \text{sumCapitalSurplus}(t) - \text{capitalExpand}(t) - \text{capitalReplenish}(t),$$

где capitalExpand — затраченный капитал, Java-переменная типа «double», рассчитывается в функции addEnterprises; capitalReplenish — недостающий капитал, Java-переменная типа «double», рассчитывается в функции replenishCapital внутри агента Enterprise; t — текущий шаг моделирования.

Функция addEnterprises обеспечивает присоединение к популяции новых агентов, т. е. расширение предпринимательской сети. На каждом шаге моде-

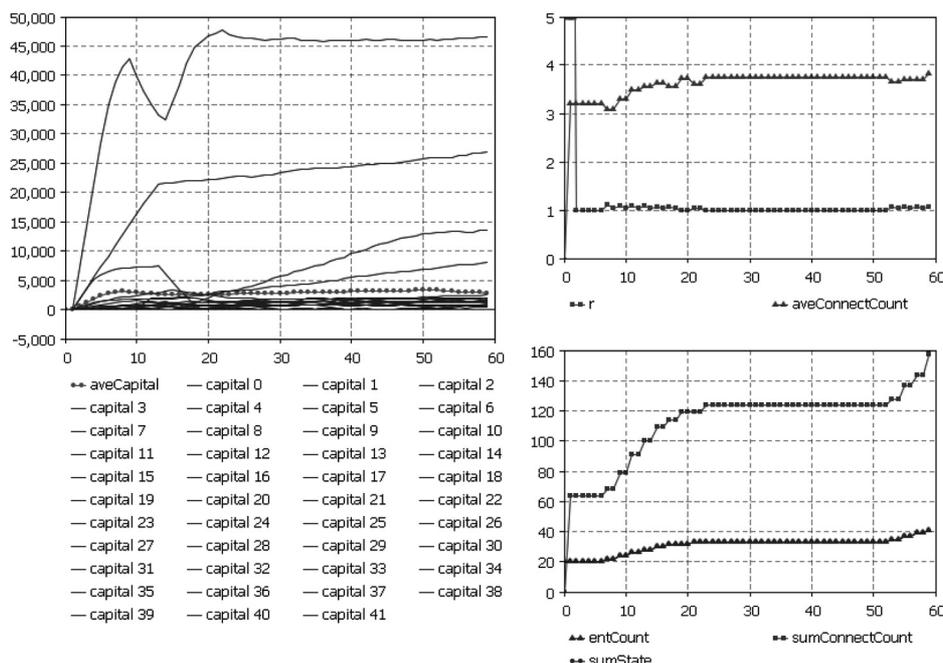


Рис. 4. Графики основных итоговых показателей

лирования, функция прибавляет к популяции одного агента, если их количество еще не достигло максимума, а сумма стартового капитала не превышает доступные средства. Также функция добавляет capital нового агента на общий график:

```

if (enterprises.size() < maxEntCount) {
    if (entCount < maxEntCount &
        sumCapitalSurplus
        >= (capitalExpend + enterprises.
            get(enterprises.size() - 1).startCapital
            + sortParameters[enterprises.size()
                [2] + capitalReplenish)
            & ((capitalUse == 1 &
                sumState == entCount) || capitalUse == 0)) {
        addEnterprises();
        entCount++;
        capitalExpend += enterprises.
            get(enterprises.size() - 1).startCapital;
        capitalPlot.
            addDataSet(enterprises.get(entCount - 1)._ds_
                capital, «capital « + (entCount - 1));
    }
}
    
```

Поскольку в начале моделирования в сеть уже входит определенное количество участников, показатели их capital также требуется выводить на график. Это делает функция Java capitalToPlot, используя системные наборы данных:

```

for (int i = 0; i < entCount; i++) {
    capitalPlot.addDataSet(enterprises.
        get(i)._ds_capital, «capital « + i);
}
    
```

Кроме общего графика капиталов всех участников сети, приложение также содержит графики основных итоговых показателей: коэффициента *r*, aveConnectCount, sumConnectCount, entCount и sumState. Графики, представленные на рис. 4, позволяют наглядно отследить изменения этих показателей в динамике на протяжении моделирования.

Третий экран соответствует системной динамике. Системная динамика отражает финансовые потоки внутри предприятия – участника сети. Поэтому переход на этот экран реализован по нажатию на пиктограмму агента. Третий экран для одного из агентов представлен на рис. 5.

Здесь расположены системно-динамические переменные, потоки и уровни, а также связи между ними. Переменная capital – капитал предприятия:

$$\text{capital}(t+1) = \text{capital}(t) + \text{receivedMoney}(t) - \text{paidMoney}(t),$$

где receivedMoney – полученные средства,

$$\text{receivedMoney}(t) = \text{receptionSum}(t) + \text{insideMoney}(t),$$

где receptionSum – вспомогательная Java-переменная; insideMoney – примененные внутренние деньги:

$$\text{insideMoney}(t) = \begin{cases} \text{если main.insideMoneyUse} == \text{true} \\ \text{если capital}(t) < 0 \\ \text{то } -\text{capital}(t) + \text{moneyLack}(t) + \text{paidMoney}(t) \\ \text{иначе } 0 \\ \text{иначе } 0 \end{cases}$$

где main.insideMoneyUse – управляющий параметр «Использование внутренних денег»; moneyLack – недостаток денежных средств,

$$\text{moneyLack}(t) = \begin{cases} \text{если receivedMoney}(t) - \text{paidMoney}(t) < 0 \\ \text{то } -(\text{receivedMoney}(t) - \text{paidMoney}(t)) \\ \text{иначе } 0 \end{cases}$$

paidMoney – выплаченные средства,

$$\text{paidMoney}(t) = \text{paymentSum}(t) + \text{moneySurplus}(t),$$

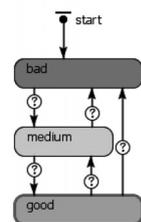
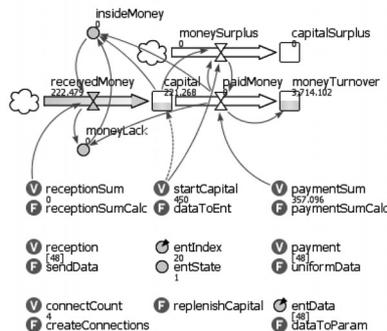


Рис. 5. Третий экран приложения — уровень системной динамики

где `paymentSum` — вспомогательная Java-переменная; `moneySurplus` — средства, потраченные на поддержку проблемных предприятий из фонда предпринимательской сети

$$\text{moneySurplus}(t) = \begin{cases} \text{если } \text{capital}(t) - \text{paidMoney}(t) > \text{startCapital} \\ \text{то } \text{capital}(t) - \text{startCapital} - \text{paidMoney}(t) \\ \text{иначе } 0 \end{cases}$$

В отличие от первой версии, при расчете `moneySurplus` учитывается `paidMoney` — сумма уже выплаченных на данном шаге средств, что исключает выход затрат за пределы доступной суммы в некоторых пограничных случаях и приближает расчеты к реальности.

Здесь `capitalSurplus` — общая сумма средств, потраченных на поддержку убыточных предприятий из фонда предпринимательской сети:

$$\begin{aligned} \text{capitalSurplus}(t+1) &= \\ &= \text{capitalSurplus}(t) + \text{moneySurplus}(t), \end{aligned}$$

`moneyTurnover` — общий оборот денежных средств агента

$$\begin{aligned} \text{moneyTurnover}(t+1) &= \\ &= \text{moneyTurnover}(t) + |\text{paidMoney}(t)|. \end{aligned}$$

На этом экране тоже присутствуют функции, что позволяет не загромождать вычисления переменных системной динамики и облегчает чтение, а значит и потенциальные доработки приложения. Данные для пошагового моделирования получены из плановой матрицы платежей путем генерации случайных чисел по закону равномерного распределения. Так при расчете значения потока `paidMoney` используются последовательно несколько функций и переменных: `dataToParam` → `entData` → `uniformData` → `paymentSumCalc` → `paymentSum`. А для потока `receivedMoney`, в свою очередь: `sendData` → `receptionSumCalc` → `receptionSum`.

Способность участников сети производить взаиморасчеты зависит от их состояния, которое, в свою очередь, зависит от капитала предприятия. Переходы между состояниями и отражение состояния цветом пиктограммы агента обеспечивает диаграмма состояний, также расположенная на этом экране. Переходы между состояниями происходят при выполнении условий. Вместо нуля используется константа равная -1 , чтобы избежать ошибочных переходов из-за особенностей округления чисел в Java:

- `badToMedium`: `capital > -1`,
- `mediumToGood`: `(capital > main.aveCapital) & (capital > -1)`,
- `mediumToBad`: `capital <= -1`,
- `goodToMedium`: `(capital <= main.aveCapital) & (capital > -1)`,
- `goodToBad`: `capital <= -1`.

Java-функция `createConnections` перед каждым шагом пересоздает связи агента с другими агентами

популяции при условии, что в исходных данных содержится ненулевая сумма платежа между ними, а состояние агента не «bad». Именно по этим связям передается сумма платежа в виде сообщения. Если связь не создана, то и сообщение передано не будет:

```
disconnectFromAll();
connectCount=0;

for(int i=0; i<main.entCount; i++){
    if(getIndex()!=i && entData[i]!=0 &&
        inState(bad)==false){
        connections.connectTo(main.
            enterprises.get(i));
        connectCount+=1;
    }
}
```

Java-функция `replenishCapital` также перед каждым шагом восполняет недостающий капитал участников сети, если это предусмотрено значением управляющего параметра `capitalUse` и на это достаточно ресурсов:

```
if(main.capitalUse==1 & main.
    sumCapitalSurplus
    >=main.capitalReplenish+abs(capital)
    +startCapital+main.capitalExpend){
    if(capital<0){
        main.capitalReplenish+=
            abs(capital)+startCapital;
        capital+=abs(capital)+
            startCapital;
    }
}
```

Выводы

Таким образом, приведенный в начале статьи алгоритм был реализован в нотации `Apylogic`, с одной стороны с использованием широких возможностей языка программирования Java, а с другой — встроенных инструментов и визуальных средств среды имитационного моделирования.

В результате получено автономное Java-приложение, которое моделирует функционирование и рост локальной предпринимательской сети. Приложение содержит три экрана: настройка управляющих параметров, агентный уровень, где сеть изображается целиком, системно-динамический уровень, отражающий каждое предприятие — участника сети изнутри. Приложение использует в качестве входных данных матрицу денежных транзакций предприятий локального сообщества. Данные задаются в файле MS Excel, их чтение и обработка происходят автоматически. Управляющие параметры дают возможность выбрать методы расширения сети и использования излишков, а также включить или исключить из системы альтернативные средства расчета. Различные сочетания управляющих параметров позволяют с помощью имитационного эксперимента оценить в динамике состояние предпринимательской сети в случае выбора той или иной стратегии. Оценка производится на основе сравнения итоговых показателей моделирования, таких, как количество связей в сети, объем капитала участников, денежный оборот и др. В приложении визуализируются состав и структура сети, а также основные итоговые показатели в виде графиков.

Обсуждение и заключение

Описанное приложение может быть полезно для поддержки принятия решений по муниципальному управлению. Лица, принимающие такие решения, часто вынуждены действовать в условиях неопределенности, что усложняет процесс и делает результат решения непредсказуемым. Использование имитационной модели как инструмента поддержки принятия решений позволяет проверить управленческие гипотезы без риска и затрат на проведение эксперимента. Реализация инструмента не просто в виде модели, а автономного компьютерного приложения расширяет возможности его применения. Разумеется, специализированное программное обеспечение часто является дорогостоящим и требует специальных знаний. Но в данной реализации оно требуется только для создания приложения, а не для его использования. Система гибких настроек минимизирует необходимость доработок при различных исходных состояниях системы и рабочих гипотезах.

Особенности конкретной реализации приложения, такие, как разделение на два уровня — агентный

и системно-динамический, позволяют посмотреть на систему с разных сторон. Системная динамика в данном случае отражает денежные потоки внутри каждого предприятия — участника сети. Агентный уровень показывает деятельность системы в целом, позволяет составить и оценить стратегию ее развития, учесть синергетический эффект, что особенно важно при управлении муниципалитетом как сложной социально-экономической системой. При этом инструмент относительно прост в использовании, не требует специальных знаний, что обеспечивает возможность применения научного подхода при принятии решений на местах, где ресурсы могут быть ограничены.

* * *

Благодарность. Настоящее исследование выполнено при финансовой поддержке гранта Российского фонда фундаментальных исследований №19-010-00974 «Экспериментальные институциональные модели автономизации финансов местных сообществ в условиях снижения доверия населения к участию в бюджетном процессе».

Список использованных источников

1. Г. Л. Багиев, Е. Г. Серова. К вопросу применения гибридного моделирования в системных исследованиях маркетингового пространства//ПСЭ. 2015. № 2 (54). С. 183-186. <https://cyberleninka.ru/article/n/k-voprosu-primeneniya-gibridnogo-modelirovaniya-v-sistemnyh-issledovaniyah-marketingovogo-prostranstva>.
2. S. C. Brailsford, L. Churilov, S.-K. Liew. Treating ailing emergency departments with simulation: An integrated perspective//Health Sciences Simulation 2003. San Diego, USA, 2003. Society for Modeling and Computer Simulation. P. 25-30. https://www.researchgate.net/publication/313092809_Treating_ailing_emergency_departments_with_simulation_An_integrated_perspective.
3. M. Jahangirian, T. Eldabi, A. Naseer et al. Simulation in manufacturing and business: A review//European Journal of Operational Research. 2010. № 203 (1). P. 1-13. <https://doi.org/10.1016/j.ejor.2009.06.004>.
4. N. Mustafee, J. H. Powell. Towards a unifying conceptual representation of hybrid simulation and hybrid systems modelling//Proceedings of the UK Operational Research Society Simulation Workshop (SW18). UKORS. 2018. P. 19-21.
5. S. C. Brailsford, T. Eldabi, M. Kunc et al. Osorio Hybrid simulation modelling in operational research: A state-of-the-art review//European Journal of Operational Research, 2019. № 278. P. 721-337. <https://doi.org/10.1016/j.ejor.2018.10.025>.
6. H. Jo, H. Lee, Y. Suh et al. A dynamic feasibility analysis of public investment projects: An integrated approach using system dynamics and agent-based modeling//International Journal of Project Management. 2015. Vol. 33. № 8. P. 1863-1876. <https://doi.org/10.1016/j.ijproman.2015.07.002>.
7. K. Kieckhäfer, T. Volling, T. S. Spengler. A Hybrid Simulation Approach for Estimating the Market Share Evolution of Electric Vehicles//Transportation Science. 2014. № 48 (4). P. 651-670. <https://doi.org/10.1287/trsc.2014.0526>.
8. G. Wallentin, C. Neuwirth. Dynamic hybrid modelling: Switching between AB and SD designs of a predator-prey model//Ecological Modelling. 2017. Vol. 345. P. 165-175. <https://doi.org/10.1016/j.ecolmodel.2016.11.007>.
9. Д. Ю. Каталевский. Системная динамика и агентное моделирование: необходимость комбинированного подхода. Сайт anylogic.ru. <https://www.anylogic.ru/resources/articles/sistemnaya-dinamika-i-agentnoe-modelirovanie-neobkhodimost-kombinirovannogo-podkhoda>.
10. А. С. Акопов. Компьютерное моделирование. М.: Издательство Урайт, 2019. 389 с.
11. A. Borshchev. Multi-method modelling: AnyLogic. In book: Discrete-event simulation and system dynamics for management decision making. 2014. P. 248-279. <https://doi.org/10.1002/9781118762745.ch12>.
12. D. B. Berg, K. A. Beklemishev MA. N. edvedev et al. Modeling of the competition life cycle using the software complex of cellular automata PyCalab//AIP Conference Proceedings. 2015. Vol. 1690. P. 030003. <https://doi.org/10.1063/1.4936702>.
13. G. K. Shevchuk, O. M. Zvereva, M. A. Medvedev. Imbalance Detection in a Manufacturing System: An Agent-Based Model Usage//AIP Conference Proceedings. 2017. Vol. 1906. P. 070013. <https://doi.org/10.1063/1.5012339>.
14. A. Kolomytseva, H. Kazakova, M. Medvedeva. Interaction Risk Assessment in Partner Entrepreneurial Networks//AIP Conference Proceedings. 2018. Vol. 1978. P. 440013. <https://doi.org/10.1063/1.5044042>.
15. А. В. Апанасенко, А. А. Паначев, Д. Б. Бергю Модельное управление ростом предпринимательской сети локального сообщества с автономизированными финансами// Современная наука: актуальные проблемы теории и практики. Серия: «Естественные и технические науки». 2021. № 2. С. 20-26. <http://www.nauteh-journal.ru/files/d4714b5d-9fee-491e-8fab-e3e24cf14b5b>.
16. D. Berg, R. Davletbaev, O. Zvereva. The model of localized business community economic development under limited financial resources: computer model and experiment// E3S Web of Conferences. 2016. Vol. 6. P. 601001. <https://doi.org/10.1051/e3sconf/20160601001>.

References

1. G. L. Bagiev, E. G. Serova. To the issue of hybrid modelling implementation in the systemic research of the market space//PSE. 2015. № 2 (54). P. 183-186. <https://cyberleninka.ru/article/n/k-voprosu-primeneniya-gibridnogo-modelirovaniya-v-sistemnyh-issledovaniyah-marketingovogo-prostranstva>. (In Russ.)
2. S. C. Brailsford, L. Churilov, S.-K. Liew. Treating ailing emergency departments with simulation: An integrated perspective//Health Sciences Simulation 2003. San Diego, USA, 2003. Society for Modeling and Computer Simulation. P. 25-30. https://www.researchgate.net/publication/313092809_Treating_ailing_emergency_departments_with_simulation_An_integrated_perspective.
3. M. Jahangirian, T. Eldabi, A. Naseer et al. Simulation in manufacturing and business: A review//European Journal of Operational Research. 2010. № 203 (1). P. 1-13. <https://doi.org/10.1016/j.ejor.2009.06.004>.
4. N. Mustafee, J. H. Powell. Towards a unifying conceptual representation of hybrid simulation and hybrid systems modelling//Proceedings of the UK Operational Research Society Simulation Workshop (SW18). UKORS. 2018. P. 19-21.
5. S. C. Brailsford, T. Eldabi, M. Kunc et al. Osorio Hybrid simulation modelling in operational research: A state-of-the-art review//European Journal of Operational Research, 2019. № 278. P. 721-337. <https://doi.org/10.1016/j.ejor.2018.10.025>.

6. H. Jo, H. Lee, Y. Suh et al. A dynamic feasibility analysis of public investment projects: An integrated approach using system dynamics and agent-based modeling//International Journal of Project Management. 2015. Vol. 33. № 8. P. 1863-1876. <https://doi.org/10.1016/j.ijproman.2015.07.002>.
7. K. Kieckhäfer, T. Volling, T. S. Spengler. A Hybrid Simulation Approach for Estimating the Market Share Evolution of Electric Vehicles//Transportation Science. 2014. № 48 (4). P. 651-670. <https://doi.org/10.1287/trsc.2014.0526>.
8. G. Wallentin, C. Neuwirth. Dynamic hybrid modelling: Switching between AB and SD designs of a predator-prey model//Ecological Modelling. 2017. Vol. 345. P. 165-175. <https://doi.org/10.1016/j.ecolmodel.2016.11.007>.
9. D. Yu. Katalievskiy. System dynamics and agent-based modeling: the need for a combined approach. Site anylogic.ru. <https://www.anylogic.ru/resources/articles/sistemnaya-dinamika-i-agentnoe-modelirovanie-neobkhodimost-kombinirovannogo-podkhoda>. (In Russ.)
10. A. S. Akopov. Computer modelling. M.: Izdatel'stvo Yurayt, 2019. 389 p. (In Russ.)
11. A. Borshchev. Multi-method modelling: AnyLogic. In book: Discrete-event simulation and system dynamics for management decision making. 2014. P. 248-279. <https://doi.org/10.1002/9781118762745.ch12>.
12. D. B. Berg, K. A. Beklemishev MA. N. edvedev et al. Modeling of the competition life cycle using the software complex of cellular automata PyCALab//AIP Conference Proceedings. 2015. Vol. 1690. P. 030003. <https://doi.org/10.1063/1.4936702>.
13. G. K. Shevchuk, O. M. Zvereva, M. A. Medvedev. Imbalance Detection in a Manufacturing System: An Agent-Based Model Usage//AIP Conference Proceedings. 2017. Vol. 1906. P. 070013. <https://doi.org/10.1063/1.5012339>.
14. A. Kolomytseva, H. Kazakova, M. Medvedeva. Interaction Risk Assessment in Partner Entrepreneurial Networks//AIP Conference Proceedings. 2018. Vol. 1978. P. 440013. <https://doi.org/10.1063/1.5044042>.
15. A. V. Apanasenko, A. A. Panachev, D. B. Berg. Model growth management of the local community entrepreneurial network with autonomous finance//Sovremennaya nauka: aktual'nye problemy teorii i praktiki. Seriya: «Estestvennye i tekhnicheskie nauki». 2021. № 2. P. 20-26. <http://www.nauteh-journal.ru/files/d4714b5d-9fee-491e-8fab-e3e24cf14b5b>. (In Russ.)
16. D. Berg, R. Davletbaev, O. Zvereva. The model of localized business community economic development under limited financial resources: computer model and experiment//E3S Web of Conferences. 2016. Vol. 6. P. 601001. <https://doi.org/10.1051/e3sconf/20160601001>.