

Имитационное моделирование процесса распространения инфекций с использованием мультиагентного подхода

В рамках поставленной задачи автор рассматривает преимущества агентного моделирования относительно альтернативных приемов имитации процессов. Разработанный программный комплекс позволяет проводить высокоэффективные вычислительные эксперименты в ходе исследований в области эпидемиологии.

Быстродействие и вычислительные мощности компьютеров позволяют использовать их как средство моделирования объектов и процессов реального мира. Математическое моделирование стало неотъемлемой частью реализации многих серьезных проектов. Оно дает возможность еще до разработки объекта оценить экономическую обоснованность и целесообразность его построения. Если ранее математическое моделирование использовалось как средство изучения различных объектов и процессов на уровне физико-химических свойств, то сейчас появляются новые возможности. Особенно незаменимо оно в тех случаях, когда эксперименты над реальными объектами затруднены или невозможны.

Одним из таких актуальных и проблемных примеров является эпидемиология. Человечество постоянно сталкивается с ранее не известными инфекциями, поэтому необходимы новые методы для уменьшения темпов их распространения. При моделировании распространения инфекций и эпидемий можно использовать несколько подходов: клеточные автоматы, дифференциальные уравнения, имитационное моделирование и т. п. Они позволяют исследовать динамику процесса распространения, а также зависимость продолжительности инфекции, количества переболевших особей и других показателей от параметров моделирования.

Принципы моделирования социальных и эпидемиологических процессов существ-

венно отличаются от моделирования в естественных науках. Здесь нет твердо установленных экспериментом и практикой зависимостей, которые всегда остаются справедливыми и не изменяются. При построении моделей таких процессов необходимо учитывать изначальную неточность задания всех данных, отсутствие четкого математического описания переменных и параметров, используемых при моделировании. Важно понимать возможность отклонения статистических данных от их реальных значений.

Стоит отметить, что существует множество разработок в области моделирования распространения инфекций и эпидемий, однако, большинство из них представляют собой системы дифференциальных уравнений, которые непригодны для практического изучения данного процесса ввиду того, что они моделируют непрерывные процессы, не учитывают индивидуальные свойства объектов, а также содержат усредненные трудно-идентифицируемые параметры. В настоящей статье предполагается использование агентного подхода для имитационного моделирования распространения инфекций.

Агентный подход в имитационном моделировании

В отличие от альтернативных моделей, агентные модели децентрализованы. Поведение агентов определяется на индивидуальном уровне, а динамика системы воз-

никает как результат деятельности многих агентов. Агентное моделирование является подходом более универсальным и мощным, так как оно позволяет учесть любые сложные структуры и поведения [3]. Другое важное преимущество агентного моделирования состоит в том, что разработка модели возможна при отсутствии знания о глобальных зависимостях. В данном случае нужно определять индивидуальную логику поведения участников процесса для того, чтобы построить агентную модель и вывести из нее глобальные поведения. Агентную модель проще поддерживать, так как уточнения обычно делаются на локальном уровне и не требуют глобальных изменений.

Мультиагентные модели — это имитационные модели, построенные для изучения интегральных характеристик системы как результата множества локальных взаимодействий ее членов (объектов).

Распространение инфекций — сложный процесс, в который вовлечены большое количество отдельных элементов (объектов), а также их устойчивые связи, взаимодействия и отношения между ними. Поведение системы складывается на основе совместного функционирования составляющих ее объектов. Основные черты процесса развития инфекции как объекта моделирования:

- большое число элементов и, как следствие, существенное количество связей между ними;
- относительная простота самих объектов и связей;
- различия в плотности связей;
- относительная простота правил взаимодействия объектов.

Поэтому для моделирования динамики распространения инфекции важным является то, что вся сложность поведения такой системы, главным образом, связана со значительным количеством простых объектов и связей. В таблице 1 показано соответствие основных характеристик модели, разрабатываемой на основе мультиагентного подхода, и процесса распространения инфекции.

Представляется логичным моделировать развитие инфекций путем создания мультиагентных моделей, у которых переходы между состояниями подчиняются некоторым правилам.

Рассмотрим фундаментальные методологические результаты компьютерного моделирования различных систем с помощью мультиагентного подхода.

Во-первых, небольшое количество логических детерминистских законов, примененных к множеству взаимосвязанных элементов, порождают сложную, хаотиче-

Таблица 1

Соответствие основных характеристик модели, разрабатываемой на основе мультиагентного подхода и процесса распространения инфекции

| Основные характеристики процесса распространения инфекции | Характеристики модели при мультиагентном подходе |
|---|--|
| Распространение инфекции представляет собой результат взаимодействия отдельных элементов (объектов) | Модель представляет собой совокупность объектов (агентов) |
| Каждый объект системы наделен определенным набором свойств | Агент обладает набором свойств, аналогичных наиболее важным свойствам реального объекта |
| Существует набор относительно простых правил, характеризующих взаимодействия объектов между собой и с окружающей средой | Существует набор правил, характеризующих взаимодействия агентов между собой и с окружением |
| Поведение системы зависит от свойств составляющих ее объектов и правил их взаимодействия | Результаты моделирования зависят от свойств агентов и правил их взаимодействия |

скую и непредсказуемую пространственную и количественную динамику согласованного коллективного поведения. Принципиально не существует аналитической формулы, с помощью которой можно было бы предсказать, за какое количество временных шагов мультиагентная модель достигнет финального состояния, какую будет иметь пространственную конфигурацию и т. д. Единственный метод изучения поведения таких моделей — их компьютерная реализация от начального до финального состояния.

Во-вторых, наблюдая за пространственными и количественными закономерностями динамики множества элементов, практически невозможно выявить простые логические детерминистские законы, которые лежат в основе функционирования модели.

В-третьих, некоторые пространственные конфигурации элементов не могут быть порождены в процессе функционирования сложной системы, а только изначально заданы.

В-четвертых, существуют правила, которые независимо от начального состояния мультиагентной модели приводят к определенным структурам, например, стационарным (устойчивым) структурам. Данные результаты имеют фундаментальное значение, поскольку они накладывают определенные методологические ограничения на принципы и методы познания законов строения и функционирования различных систем и поведения индивидов.

Платформы для разработки агентно-ориентированных систем

Кратко рассмотрим основные средства для разработки мультиагентных систем.

Для построения интеллектуальных агентов может использоваться платформа ABLE. Она распространяется с открытым исходным кодом, способна работать на системах Windows, Unix, OS/2 и требует знания языка программирования Able Rule Language.

Создание агентных моделей для прогнозирования цен акций и других ценных бумаг позволяет производить Adaptive Modeler.

Способна работать на ОС с установленным .Net Framework 2.0 и выше, при этом не требуется опыт программирования. Бесплатно доступна только ознакомительная версия.

Для построения агентных систем общего назначения создана платформа AgentBuilder, требующая знания языков KQML, Java, C, C++. Является платной.

Моделирование в социальных и технических науках позволяет проводить платформа NetLogo, которая распространяется бесплатно (но без исходного кода), способна работать на любой платформе, где есть Java, обязательно знание языка программирования NetLogo.

Мощным инструментом для моделирования является платформа JADE, которая также требует Java и распространяется по лицензии LGPL.

Российская разработка, позволяющая использовать все виды имитационного моделирования, — платформа AnyLogic. Следует отметить, что она является проприетарным программным обеспечением и требует знания языков Java и UML-RT.

Итак, для разработки мультиагентных моделей необходимо знание определенных языков программирования для проведения вычислительных экспериментов. В случае моделирования распространения инфекций целесообразно создание симулятора, который может быть применен эпидемиологами для изучения самого процесса распространения без детального погружения в моделирование.

В данной работе предполагается использование свободной среды разработки программного обеспечения Lazarus, основанной на компиляторе Free Pascal.

Основные идеи мультиагентного моделирования распространения инфекции

Для целей моделирования приняты следующие допущения:

1. Моделирование осуществляется на ограниченной территории, где существу-

ет вероятность взаимодействия агента с любым другим из системы.

2. Время в модели дискретно. Единицей времени является одна итерация. В реальном времени она может представлять собой один день, месяц, год и т. п. в зависимости от значений параметров модели. Отсчет времени начинается с нуля итераций. Шаг времени равен единице.

3. Максимальная продолжительность жизни агента задается количеством итераций, по истечении которых агент прекращает свое существование. Максимальная продолжительность жизни может быть изменена в процессе моделирования, в случае инфицирования агента.

4. Все агенты разбиваются на типы, для каждого типа объектов существует система правил, взаимодействие по которым определяет характер динамики распространения инфекции.

5. Взаимодействие одновременно возможно только между двумя агентами системы. То есть невозможно такое взаимодействие, в результате которого изменяются параметры нескольких агентов.

6. При инфицировании агента и изменении его типа в момент времени t способность к передаче инфекции возникает в момент времени $t+1$.

В начальный момент времени имеется M — множество объектов различного типа:

$$M = \{C_1, C_2, \dots, C_p\},$$

где p — количество типов объектов;

$$C_k = \{c_1^k, c_2^k, \dots, c_{n_k}^k\}, \quad k = \overline{1, p},$$

где C_k — множество объектов типа k ;

n_k — общее количество объектов типа k в начальный момент времени.

Параметры объектов типа k задаются в соответствии с выражением

$$c_i^k \mapsto \{s_i^k, a_i^k, am_i^k, w_i^k\}, \quad i = \overline{1, n_k},$$

где s_i^k — пол объектов;

a_i^k — возраст объектов в начальный момент времени;

am_i^k — максимальная продолжительность жизни объектов типа k ;

w_i^k — количество взаимодействий объекта с другими объектами.

Функционирование объектов подчинено системе правил $U = \{U_1, U_2, U_3, \dots, U_n\}$ [2]. Согласно этим правилам, инфицированный агент при взаимодействии непрерывно генерирует сигнал инфекции. Если сигнал инфекции получает здоровый агент, то с некоторой вероятностью он изменяет свое состояние на «инфицирован». При этом у обладающего иммунитетом агента такая вероятность заведомо меньше, чем у агента, подверженного заражению.

Возникновение i -го объекта типа k в момент времени $t+1$ и его гибель определяют выражениями

$$\begin{aligned} (C_k)_t &\rightarrow (C_k + \{c_i^k\})_{t+1}; \\ (C_k)_t &\rightarrow (C_k - \{c_i^k\})_{t+1}. \end{aligned}$$

Изменение параметра a_k — возраста объекта типа k за единицу времени происходит согласно системе:

$$\begin{aligned} &\left\{ \begin{aligned} (c_i^k)_{t-1} &\rightarrow (c_i^k)_t : (a_i^k)_t = (a_i^k)_{t-1} + 1 \\ (a_i^k)_t &\geq am_i^k \Rightarrow (C_k)_t \rightarrow (C_k - \{c_i^k\})_{t+1} \end{aligned} \right\}, \\ &k = \overline{1, p}, \quad i = \overline{1, n_k}. \end{aligned}$$

По второму правилу системы необходимо исключить объект из множества объектов типа k , если отведенный ему максимальный срок жизни уже истек.

Максимальный срок жизни объектов типа k может изменяться на заданное количество итераций h_i , определяющее максимальную продолжительность жизни после изменения типа в результате взаимодействия объектов разных типов:

$$\begin{aligned} &\{c_i^k \in C_k \mid k_{t+1} \neq k_t\} \Rightarrow \\ &\Rightarrow (am_i^k)_{t+1} = \begin{cases} (a_i^k)_t + h_i, & (a_i^k)_t + h_i < am_i^k, \\ am_i^k, & (a_i^k)_t + h_i \geq am_i^k \end{cases} \end{aligned}$$

Правила взаимодействия агентов позволяют производить моделирование «снизу вверх», т. е. исследовать динамику сложной системы, оказывая влияние на составляющие ее объекты.

Программный комплекс для реализации модели

Для реализации программного комплекса были применены методы объектно-ориентированного программирования [1]. Программный комплекс создан в среде Lazarus. Требования к операционной системе: Linux, Microsoft Windows, Mac OS X, FreeBSD, OS/2. Необходимо отметить, что для моделирования большого количества объектов требуется платформа x64.

Для хранения объектов использовался класс *People* (см. листинг 1).

При создании объекта определяются его характеристики: тип, возраст, максималь-

ная продолжительность жизни, пол, количество контактов и вероятность инфицирования.

Функция *kinf* позволяет определить количество объектов заданного типа. Процедуры *newinf* и *newdead* используются в случае инфицирования и смерти агента соответственно.

Для проведения вычислительных экспериментов над выбранным исходным множеством агентов, чтобы каждый раз не вводить параметры объектов, предусмотрены процедуры экспорта и импорта (см. листинг 2).

Также для быстрого анализа множества, экспортированного в файл, доступна специальная процедура (см. листинг 3).

Визуально рабочая область программы разделена на две зоны, каждая из которых служит для представления различной информации. На рисунке 1 (см. стр. 86) показан общий вид программного комплек-

Листинг 1. Описание класса *People*

```

People = class
  private
    sex: boolean;
    age: byte;
    Mage: byte;
    typeobject: byte;
    contact: array[1..5] of integer;
    pinf: array[1..5] of byte;
    dead: boolean;
  public
    Constructor create(qtype, qage, qmaxage: Integer; qsex, qdead: boolean; qcontact:
    Tpf; qpinf: Tpf);
    function kinf(qtype: byte): Integer;
    procedure newinf();
    procedure newdead();
    procedure counter(x : longint);
  end;

Constructor People.create(qtype, qage, qmaxage: Integer; qsex, qdead: boolean;
qcontact: Tpf; qpinf: Tpf);
begin
  typeobject := qtype;
  age := qage;
  Mage := qmaxage;
  sex := qsex;
  dead := qdead;
  contact := qcontact;
  pinf := qpinf;
end;

```

Листинг 2. Процедуры экспорта и импорта параметров объектов

```

procedure TMainForm.Export1Click(Sender: TObject);
var
  expf: TextFile;
typeobject, ageobject, sexobject, kolinf: Integer;
begin
  try
    if not(datamodule1.SaveDialog1.Execute) then Abort;
    AssignFile(expf, datamodule1.SaveDialog1.FileName);
    Rewrite(expf);
    Writeln(expf, index);
    Writeln(expf, kt);
    Writeln(expf, kpp);
    for i:=1 to index do
      begin
        Writeln(expf, allagent[i].typeobject);
        Writeln(expf, allagent[i].age);
        Writeln(expf, allagent[i].Mage);
        Writeln(expf, integer(allagent[i].sex));
        Writeln(expf, integer(allagent[i].dead));
        for j:=1 to high(allagent[i].contact) do
          Writeln(expf, allagent[i].contact[j]);
        for j:=1 to high(allagent[i].pinf) do
          Writeln(expf, allagent[i].pinf[j]);
        end;
      end;
    closefile(expf);
    messagebox(0, RusText('Экспорт данных завершен'), RusText('Экспорт'), MB_OK);
  except
    messagebox(0, RusText('Экспорт данных невозможен. Возможно, отсутствуют права на запись в файл или нет данных для экспорта'), RusText('Ошибка экспорта данных'), MB_OK+MB_ICONERROR);
  end;
end;

procedure TMainForm.Import1Click(Sender: TObject);
var
  impf: textfile;
countobject: LongInt;
typeobject, ageobject, maxageobject, sexobject, deadobject: Integer;
begin
  try
    if not(datamodule1.OpenDialog1.Execute) then Abort;
    AssignFile(impf, datamodule1.OpenDialog1.FileName);
    Reset(impf);
    Readln(impf, countobject);
    Readln(impf, kt);
    Readln(impf, kpp);
    index := 0;
    SetLength(allagent, countobject+1);
    for i:=1 to countobject do
      begin
        inc(index);
        Readln(impf, typeobject);
        Readln(impf, ageobject);
        Readln(impf, maxageobject);
        Readln(impf, sexobject);
        Readln(impf, deadobject);
      end;
    end;
  end;
end;

```

Продолжение листинга 2

```

    for j:=1 to high(kv) do
        Readln(impf,kv[j]);
    for j:=1 to high(vz) do
        Readln(impf,vz[j]);
        agent := People.create(typeobject, ageobject, maxageobject, bool(sexobject),
bool(deadobject), kv, vz);
        allagent[index] := agent;
    end;
    Edit7.Text:= inttostr(kt);
    Edit2.Text:= inttostr(kpp);
    button7.click;
    button2.click;
    closefile(impf);
    messagebox(0, RusText('Импорт данных завершен. Импортировано объектов:
'+IntToStr(countobject)), RusText('Импорт'), MB_OK);
except
    messagebox(0, RusText('Импорт данных невозможен. Проверьте формат импортируемого
файла'), RusText('Ошибка импорта данных'), MB_OK+MB_ICONERROR);
end;
end;
end;

```

Листинг 3. Процедура анализа экспортированного множества

```

procedure TMainForm.MenuItem3Click(Sender: TObject);
var
    impf: textfile;
    tempmas: array [1..10] of byte;
    countobject: LongInt;
    typeobject, ageobject, maxageobject, sexobject, deadobject,
    ksexM, ksexW, kdead: Integer;
    msg: string;
begin
    try
        if not(datamodule1.OpenDialog1.Execute) then Abort;
        AssignFile(impf,datamodule1.OpenDialog1.FileName);
        Reset(impf);
        ksexM:=0;
        ksexW:=0;
        kdead:=0;
        for i:=1 to 10 do
            tempmas[i] := 0;
            Readln(impf,countobject);
            Readln(impf,kt);
            Readln(impf,kpp);
            index := 0;
            SetLength(allagent,countobject+1);
            for i:=1 to countobject do
                begin
                    inc(index);
                    Readln(impf,typeobject);
                    Readln(impf,ageobject);
                    Readln(impf,maxageobject);
                    Readln(impf,sexobject);
                    Readln(impf,deadobject);
                    for j:=1 to high(kv) do
                        Readln(impf,kv[j]);
                    for j:=1 to high(vz) do

```

Продолжение листинга 3

```

Readln(impf,vz[j]);
if bool(sexobject) then inc(ksexM) else inc(ksexW);
if bool(deadobject) then inc(kdead);
inc(tempmas[typeobject]);
end;
msg := '';
for i:=1 to kt do
msg := msg+#10#13+'Количество объектов типа TYPE' + inttostr(i) + ': ' +
inttostr(tempmas[i]);
closefile(impf);
messagebox(0, RusText('Анализ данных завершен. Всего объектов: '+IntToStr(countobject)
+ #10#13 + 'Мужчин: '+IntToStr(ksexM) + #10#13 + 'Женщин: ' + IntToStr(ksexW) + msg +
#10#13 + 'Погибших: ' + IntToStr(kdead)), RusText('Анализ данных'), MB_OK);
except
messagebox(0, RusText('Анализ данных невозможен. Проверьте формат файла данных'),
RusText('Ошибка анализа данных'), MB_OK+MB_ICONERROR);
end;
end;
end;

```

са. Левая панель предназначена для задания количества типов объектов, путей передачи, а также изменения каталога хранения данных. Он содержит промежуточные и конечные результаты процесса моделирования, которые затем можно анализировать, используя функцию анализа данных. Здесь же задается количество моделируе-

мых итераций. Самая верхняя строка левой панели отображает общее количество заданных объектов. Для отображения актуального количества объектов необходимо кликнуть левой кнопкой мыши по надписи **«Обновить данные»**. После ввода количества типов объектов и путей передачи информации становится активной правая панель.

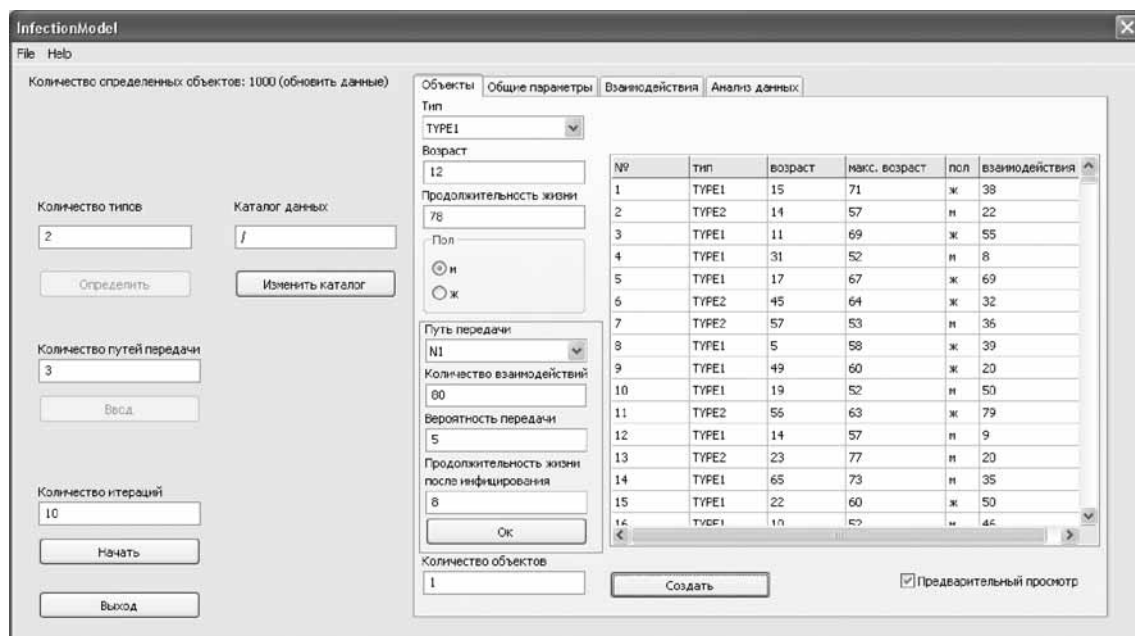


Рис. 1. Общий вид программного комплекса

Она разделена на 4 вкладки: «Объекты», «Общие параметры», «Взаимодействия» и «Анализ данных».

Вкладка **«Объекты»** позволяет добавлять в систему новых агентов, указывать индивидуальные свойства и предварительно просматривать их. Здесь задаются следующие свойства: тип объекта (из выпадающего списка), возраст, продолжительность жизни, пол, количество взаимодействий и вероятность инфицирования по каждому пути передачи, а также продолжительность жизни после инфицирования.

Вкладка **«Общие параметры»** содержит две области (рис. 2). В левой части задаются общие параметры инфекции: количество взаимодействий, вероятность инфицирования и продолжительность жизни инфицированного агента, которые применяются к тем объектам, в чьих индивидуальных свойствах они отсутствуют. Правая часть позволяет определить процессы иммиграции и эмиграции населения. На любой итерации есть возможность задать количество эмигрантов и иммигрантов. Необходимо отметить, что моделируется только

процесс распространения инфекции, общая динамика населения должна задаваться согласно статистическим данным. Для определения процесса миграции на соответствующей итерации следует выбрать направление движения агентов: удаление из системы (эмигранты) или добавление в нее новых агентов (иммигранты), а затем задать параметры объектов и их количество.

С помощью вкладки **«Взаимодействия»** определяют, какие из существующих взаимодействий в системе приводят к изменению типа объекта. В левой части выбираются типы взаимодействующих объектов, а в правой — результат их взаимодействия (рис. 3). Как отмечалось ранее, взаимодействие возможно одновременно только между двумя объектами системы, однако, агент за одну итерацию совершает несколько таких взаимодействий, в результате чего может изменить состояние нескольких других агентов.

Вкладка **«Анализ данных»** предназначена для исследования процесса распространения инфекции и позволяет определить

Рис. 2. Вкладка «Общие параметры»

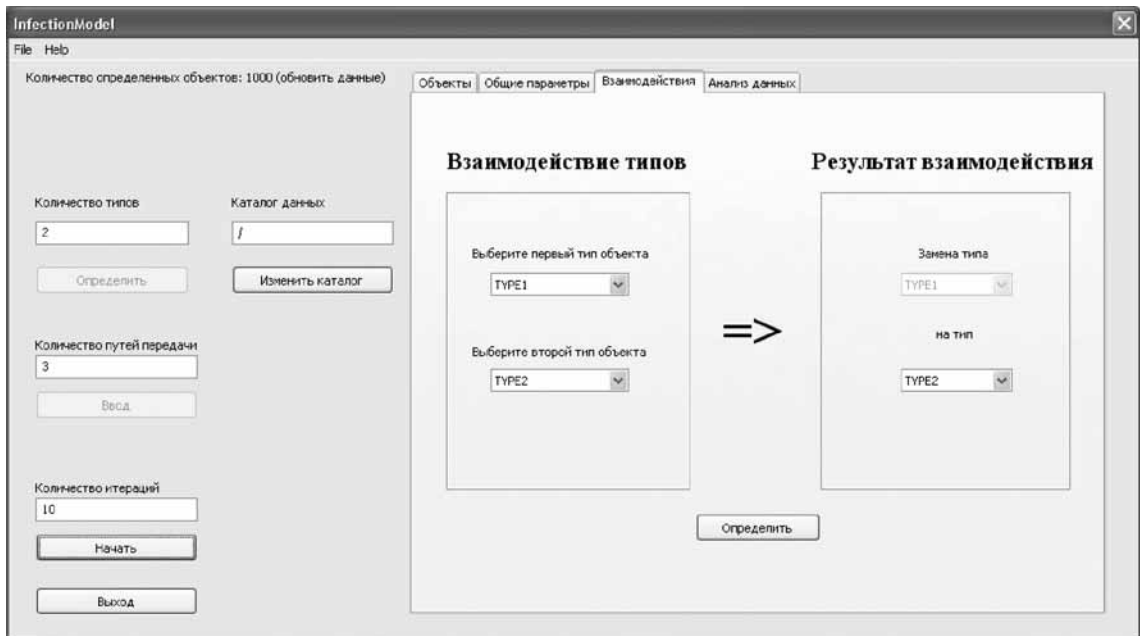


Рис. 3. Вкладка «Взаимодействия»

количество агентов с заданным набором параметров (рис. 4). В левой панели нужно выбрать параметры объектов: пол, возраст и тип. Необходимо отметить, что воз-

можен анализ множества объектов, которое непосредственно определено в программе, и множества, записанного в файл. Для этого в правой панели выбирают соответствующую

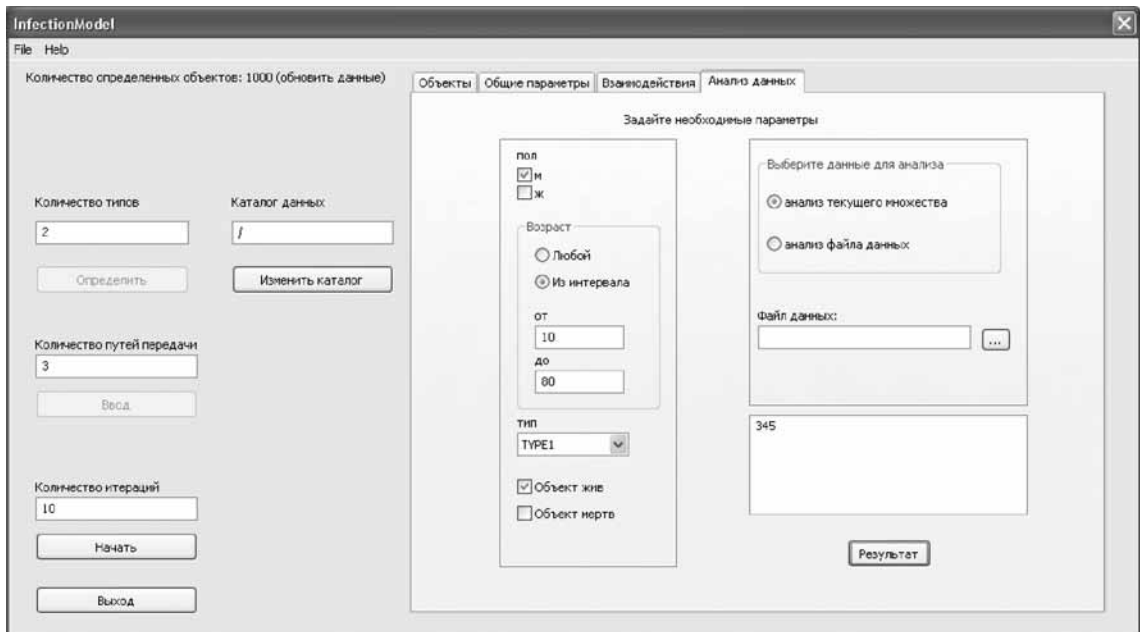


Рис. 4. Вкладка «Анализ данных»

щий переключатель и в случае, если выбран анализ файла данных, указывается полный путь к файлу.

Кроме файлов, представляющих собой промежуточные результаты моделирования, существует возможность выгрузки текущего множества агентов, например, для проведения над ним нескольких вычислительных экспериментов. С этой целью используется пункт **Export** меню **File**. В результате появится диалоговое окно, в котором необходимо указать каталог хранения файла и его имя, а затем кликнуть мышью по кнопке «Сохранить». Также возможно импортирование заданного множества из файла с использованием пункта **Import** меню **File**, для чего нужно выбрать файл данных и кликнуть мышью по кнопке **Открыть**.

У программного комплекса свой внутренний формат. *.dat* для хранения данных, содержащий общее количество определенных объектов, количество типов, а также путей передачи инфекции и информации о каждом отдельном объекте (тип, возраст, максимальная продолжительность жизни и т. д.). В случае необходимости возможно создание такого файла с помощью иных средств при соблюдении строгой последовательности записи данных в файл. При попытке импорта файла иного формата возникает сообщение об ошибке импортирования данных.

Меню **File** содержит также пункт **Analysis**, который позволяет провести быстрый анализ множества, экспортированного в файл, согласно описанной выше процедуре.

Заключение

Разработанный программный комплекс, предназначенный для анализа и проведения вычислительных экспериментов по мультиагентной модели распространения инфекции, позволяет задавать индивидуальные свойства и параметры объектов. Многие приемы, используемые при применении мультиагентного подхода, в настоящее время проработаны в недостаточной степени. В частности,

не решены задачи параметрической идентификации указанных моделей, нахождения необходимого числа агентов, обеспечивающих репрезентативность вычислительных экспериментов, проверки мультиагентных моделей на адекватность. По этой причине разработка новых математических моделей, алгоритмов и программ, предназначенных для применения мультиагентного подхода при анализе различных систем, представляет собой актуальную научную задачу.

Разработанный автором программный комплекс обладает кросс-платформенностью и имеет дружественный интерфейс, что обеспечивает хорошую переносимость и позволяет использовать его для анализа процессов распространения инфекций специалистами в области эпидемиологии. При этом пользователи необязательно должны являться специалистами в области математического моделирования, что существенно расширяет практическую востребованность разработанного программного средства.

Список литературы

1. Алексеев Е. Р., Чеснокова О. В., Кучер Т. В. Free Pascal и Lazarus. Учебник по программированию. М.: ALT Linux; Издательский дом «ДМК-пресс», 2010.
2. Арзамасцев А. А., Улыбин А. В. Имитационное моделирование развития инфекции с использованием агентного подхода // Вестник Тамбовского университета. Серия: Естественные и технические науки. 2010. Т. 15. № 2.
3. Кельтон Д. В., Лоу А. М. Имитационное моделирование. СПб.: Питер, 2004.
4. Емельянов А. А., Власова Е. А., Дума Р. В., Емельянова Н. З. Компьютерная имитация экономических процессов / Под. ред. А. А. Емельянова. М.: Маркет ДС, 2010.
5. Карпов Ю. Г. Имитационное моделирование систем. Введение в моделирование с AnyLogic 5. СПб.: БХВ-Петербург, 2005.
6. D'Inverno M., Luck M. Understanding Agent System. New York: Springer-Verlag, 2001.
7. Weiss G. Multiagent Systems: A Modern Approach to Distributed Artificial Intelligence. Cambridge: MIT Press, 1999.