УДК: 004.942 **ГРНТИ:** 20.23.17

ПОСТРОЕНИЕ МОДЕЛИ ПРЕДОСТАВЛЕНИЯ ТЕХНИЧЕСКОЙ ПОДДЕРЖКИ ПОЛЬЗОВАТЕЛЕЙ НА ЯЗЫКЕ МОДЕЛИРОВАНИЯ ANYLOGIC

А. Р. Срульдинов, С. А. Варламова*
Пермский национальный исследовательский университет
Россия, 618404, г. Березники, ул. Тельмана, 7
* email: varlamovasa@mail.ru

Описаны основные принципы разработки агентной модели на языке AnyLogic. Обоснованы преимущества имитационного моделирования для широкого круга задач. Разработаны структурная и агентная модели технической поддержки пользователей. Описаны элементы модели и значения ее параметров. Произведен анализ результатов моделирования.

Ключевые слова: *агентная модель, техническая поддержка, AnyLogic.*

DESIGN OF TECHNICAL SUPPORT MODEL BY ANYLOGIC LANGUAGE

A. R. Sruldinov, S. A. Varlamova*
Perm national polytechnic research university
7 Telmana St., 618404, Berezniki, Russia
* email: varlamovasa@mail.ru

Basic principles of agent model design by AnyLogic are described. Features of queue modeling for wide specter of tasks are shown. A structural and queue models of technical support service are designed. All elements of queue model are circumscribed, parameters of the model are based on a real object observing. An analysis of modeling results is given.

Keywords: *agent model, support service, AnyLogic.*

Современный этап развития человечества отличается тем, что на смену века энергетики приходит век информатики. Происходит интенсивное внедрение новых информационных технологий во все сферы человеческой деятельности.

Появление новейших информационных технологий увеличивает не только возможности моделирующих систем, но и позволяет применять большее многообразие моделей и способов их реализации.

В настоящее время подготовка управленческих решений требует принятия во внимание большого числа различных факторов, влияющих на динамику исследуемых систем. Лицам, принимающим решения, необходимо анализировать сотни различных сценариев (вариантов решений), что обуславливает необходимость разработки имитационных моделей различного типа, интеграции этих моделей с базами и хранилищами данных, создания оптимизационных модулей для имитационных моделей.

Суть имитационного моделирования заключается в компьютерной реализации математической модели изучаемой системы для использования в целях симуляции (имитации) поведения реальной системы.

В настоящее время сфера применения имитационного моделирования практически не ограничена. Имитационные модели разрабатываются для изучения поведения отдельных предприятий и корпораций, исследования рыночного равновесия, оценки последствий государственного регулирования в экономике, оптимального управления логистическими системами, изучения эпидемиологии (прогнозирования динамики распространения заболеваний) и решения других практических задач.

В данной статье попробуем с помощью компьютерной имитационной модели описать процесс предоставления технической поддержки пользователям.

В отдел технической поддержки пользователей поступают заявки n типов с вероятностями p1, p2, …, pn соответственно. Интервалы времени Tп между двумя очередными поступлениями одного типа заявок случайные. Каждый любой тип заявки может требовать одного из a1, a2, …, ak видов работ с вероятностями pa1, pa2, …, pak соответственно.

В отделе работают n1, n2, …, nn специалистов для выполнения заявок каждого типа соответственно. Специалист n1 выполняет заявки первого типа, если таких заявок нет и остальные специалисты заняты, то специалист n1 выполняет заявки остальных типов, при этом поступающие заявки первого типа ожидают его освобождения. Аналогичные обязанности и у остальных специалистов, кроме специалиста nn, он выполняет заявки одного n-го типа, в нашем случае это разработчик.

Время выполнения заявки случайное, не зависит от специалиста, а зависит только от поставленной задачи: T11,…, T1n – для заявок первого типа, T21,…, T2n – для заявок второго типа, Tn1,…, Tnn – для заявок n-го типа.

Прием и распределение заявок между специалистами осуществляется d диспетчерами. Время, затрачиваемое одним диспетчером на одну заявку, T1, случайное. Диспетчерами не принимаются к ремонту q заявок всех типов.

Интервалы времени между поступлениями заявок, и время выполнения заявок распределены по экспоненциальному закону.

Представим отдел технической поддержки как СМО (рис. 1).

Отдел технической поддержки представляет собой многофазную многоканальную систему массового обслуживания разомкнутого типа с отказами.

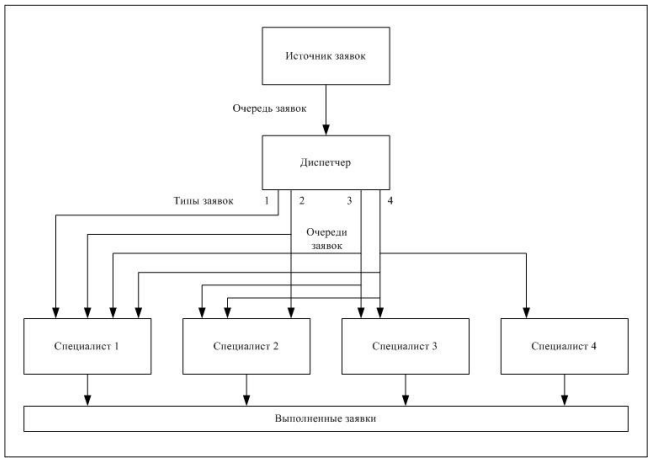

Рис. 1. Отдел технической поддержки как СМО

Заявки должны иметь следующие параметры (поля):

- типЗ — код типа заявки;
- видР — вид работы;
- времяР — время выполнения одного вида работы;

Как говорилось выше, интервалы между соседними заявками подчинены экспоненциальному закону. Принято, что за время Tp от каждого источника поступает одна заявка, тогда средний интервал поступления заявок равен Tp/n, поэтому вместо n объектов имитации источников заявок будем использовать один.

Код типа заявки определяется в виде чисел 1, 2, 3, 4, так как n=4. Код вида работы определяется также числами 1...3 соответственно. Для этого используются следующие исходные данные:

p1 ... p4 — вероятности поступления заявок 1...4 типов соответственно;

p11 ... p43 — вероятности поступления заявок 1...4 типов с видами 1...3 работ соответственно.

Коды типа заявки и вида работ записываются в поля типЗ и видР соответственно. По этим кодам определяется среднее время вида работы и заносится в поле времяР.

В процессе выполнения модели накапливаются следующие статистические данные:

постЗаявТип1 ... постЗаявТип4, постЗаявТип — количество поступивших заявок 1...4 типов и заявок всех типов;

выпЗаявТип1 ... выпЗаявТип4, выпЗаявТип — количество выполненных заявок 1...4 типов и заявок всех типов;

выпРабВида11 ... выпРабВида43 — количество выполненных заявок 1...4 типов с видами 1...3 работ соответственно.

Поскольку эти данные накапливаются за все прогоны модели, то для получения средних значений они делятся на количество прогонов колПрог. Например, выпЗаявТип1=выпЗаявТип1/ колПрог.

По этим же статистическим данным рассчитываются:

верВыпЗаяв1 ... верВыпЗаяв4, верВыпЗаяв — вероятности выполнения заявок 1...4 типов и заявок всех типов.

Например, верВыпЗаяв1=выпЗаявТип1/постЗаявТип1.

Модель в AnyLogic
Ввод исходных данных и вывод результатов

Для ввода исходных данных нам понадобятся несколько элементов «Параметр». «Параметр» используется для задания статических характеристик агента.

Для вывода результатов будет использоваться элемент «Переменная». Она предназначена для моделирования изменяющихся характеристик агента или для хранения результатов работы модели (Рис. 2).

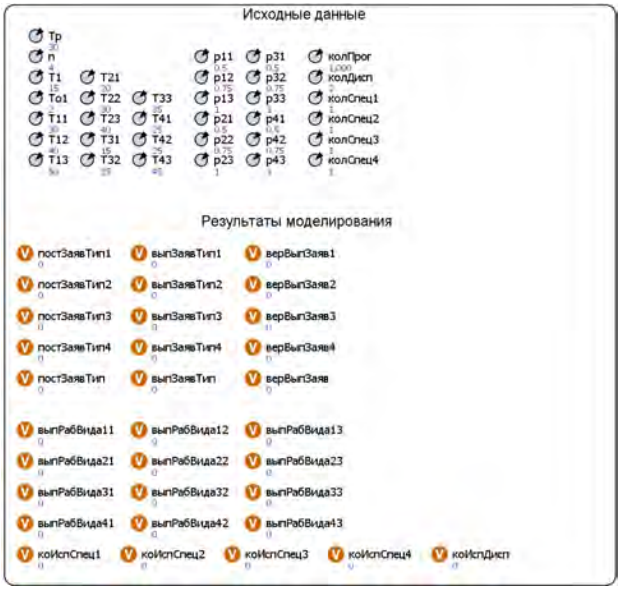

Рис. 2. Размещение элементов Параметр и Переменная

Таблица 1

Элементы и их свойства			
Параметр		Параметр	
Имя	Значение по умолчанию	Имя	Значение по умолчанию
Tp n	30 4	T1 To1	15 2
T11 T12 T13	30 40 50	p11 p12 p13	0.50 0.75 1.00
T21 T22 T23	20 30 40	p21 p22 p23	0.50 0.75 1.00
T31 T32 T33	15 25 35	p31 p32 p33	0.50 0.75 1.00
T41 T42 T43	25 35 45	p41 p42 p43	0.50 0.75 1.00
колПрог	1000	колДисп	2
колСпец1 колСпец2	1 1	колСпец3 колСпец4	1 1

В таблице (Табл. 1) указаны значения параметров элементов, полученные по результатам наблюдений за реальным объектом и путем анализа его работы.

Вся модель будет разбита на 4 сегмента:

- источник заявок;
- диспетчеры;
- специалисты;
- учёт выполненных заявок.

Далее остановимся конкретнее на каждом из сегментов.

Сегмент Источник заявок состоит из одного элемента Source, который создает агентов и используется в качестве начальной точки потока агентов.

Так же в этом сегменте присутствует Java класс с названием Заявка. Он содержит следующие поля Java класса: типЗ, видР, времяР.

Java-кодом в поля entity.тип3 и entity.видР заносятся равномерно распределённые случайные числа. Они нужны далее для розыгрыша кодов типов заявок и кодов видов ремонта.

Сегмент Диспетчеры предназначен для распределения по специалистам заявок согласно их типам и видам работ в зависимости от занятости мастеров в текущий момент времени.

Данный сегмент содержит следующие объекты:

SelectOutput5 – объект направляет входящих агентов в один из пяти выходных портов в зависимости от выполнения заданных (детерминистических или заданных с помощью вероятностей) условий.

SelectOutput – направляет входящих агентов в один из двух выходных портов в зависимости от выполнения заданного (детерминистического или заданного с помощью вероятностей) условия.

Queue – моделирует очередь агентов, ожидающих приема объектами, следующими за данным в потоковой диаграмме, или же хранилище агентов общего назначения.

Delay – задерживает агентов на заданный период времени. Время задержки вычисляется динамически, может быть случайным, зависеть от текущего агента или от каких-то других условий.

Sink – используется в качестве конечной точки потока агентов.

Объектом типЗаявки (SelectOutput5) разыгрывается код типа заявки. Например, в поступившей заявке agent.тип3=0.723. Проверяется условие 0: agent.тип3=0.723<=p1=0.5. Условие 0 не выполняется. Тогда проверяется условие 1: agent.тип3= 0.723 <=p2=0.75. Условие 1 выполняется. Заявка пропускается на выход 1.

С выходов 0...3 объекта типЗаявки заявки поступают в очДисп (queue) с максимальной вместимостью, а затем в объект дисп (delay), имитирующий время работы одного из диспетчеров с одной заявкой.

Объект отказ (selectOutput) предназначен для розыгрыша отказа в принятии заявки с вероятностью q = 2 %. Заявки, получившие отказ, уничтожаются объектом sink.

Принятые к выполнению заявки распределяются по типам объектом поТипамЗаяв (SelectOutput5), с выходов 0...3 этого объекта заявки поступают на объекты видРабЗаяв1... видРабЗаяв4 соответственно. Аналогичным образом как объектом типЗаявки этими объектами разыгрываются для заявок 1...4 типов коды видов 1...3 ремонтов.

Сегмент Специалисты предназначен для имитации ожидания освобождения специалистов, непосредственно времени выполнения соответствующего вида работ и отправки выполненной заявки в сегмент учёта.

Данный сегмент содержит объекты queue и delay, которые были описаны выше.

С выходов 1...3 объекта видРабЗаяв1 заявки сразу поступают в объект очСпец1 (queue).

С выходов объектов видРабЗаяв2...видРабЗаяв4 заявки поступают на объекты свобСпец1_2, свобСпец1_3, своб-Спец1_4 соответственно. В принятых именах первая цифра означает группу мастеров, а вторая — тип заявки. Этими объектами проверяются условия. Например, объектом свобМастер1_3 проверяется условие:

(очСпец1.size()==0)&&

(спец1.size()<колСпец1)&&(спец3.size()!=0)

Пуста ли очередь специалиста 1? И свободен ли специалист 1? И занят ли специалист 3 группы? Если сложное условие, состоящее из трёх простых условий, выполняется, заявка с выхода true объекта своб Спец1_3 на объект спец1.

Аналогичные проверки осуществляются в объектах своб Спец1_3, своб Спец1_4.

Если условие не выполняется, то заявка с выходов false поступает в очередь очСпец2...очСпец4 соответственно.

Объекты своб Спец2_3, своб Спец2_4 проверяют возможности в текущий момент времени выполнения заявок 3 и 4 типов специалистом 2.

Объект своб Спец3_4 проверяет возможность выполнения заявок 4 типа специалистом 3.

Сегмент Учёт выполненных заявок предназначен для учёта количества выполненных заявок по типам и видам работ, а также для определения вероятности выполнения заявок в целом.

Сегмент построен на объектах selectOutput5 и объекте sink.

Объект поТипамЗаяв1 осуществляет разделение и учёт выполненных заявок по типам, а также рассчитывает вероятности выполнения заявок каждого типа. В количество поступивших заявок, а, следовательно, и в расчёт вероятностей входят и те заявки, которым отказано в обслуживании.

Объекты выпРабЗаяв1...выпРабЗаяв4 учитывают количество видов ремонтов, выполненных по заявкам каждого типа.

Объект sink1 уничтожает поступающие заявки. Введён-

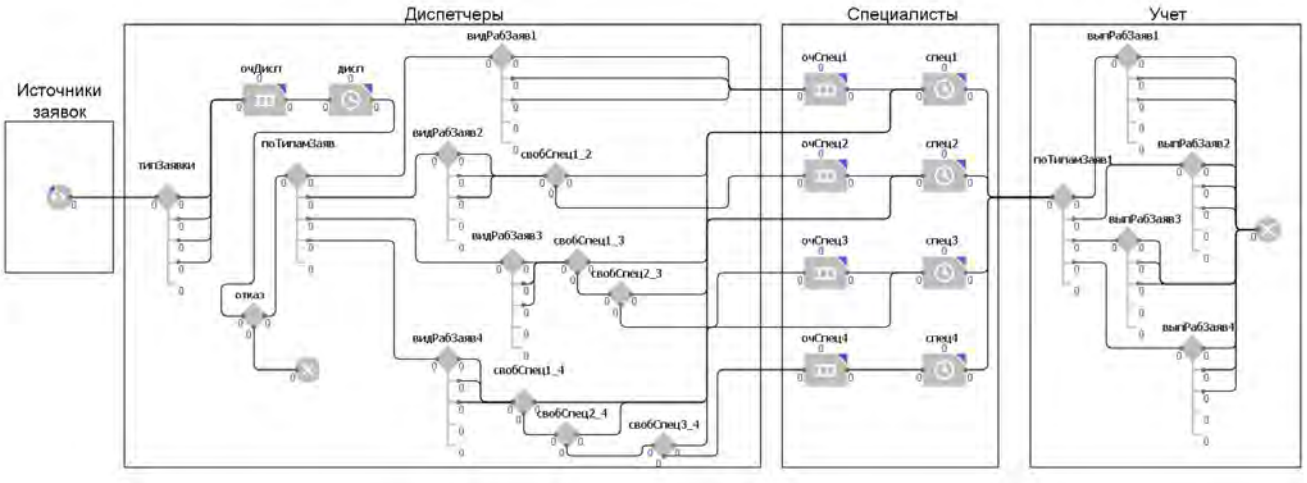

Рис. 3 Вид готовой модели на AnyLogic

ный в поле Действие при входе код рассчитывает вероятность выполнения всех заявок.

После проведения выше описанных действий готовая модель выглядит следующим образом (Рис. 3).

В результате исследования работы модели получили следующие данные, заявок 1 типа выполнено 9853, с вероятностью выполнения 0,789, всего выполнено заявок всех типов 46300 с вероятностью 0,927. Остальные данные работы модели приведены в таблице (Табл. 2).

Таблица 2

Показатели	
Выполнено заявок типа 1	9853
работ вида 11	4958
работ вида 12	2481
работ вида 13	2414
Выполнено заявок типа 2	12089
работ вида 21	6028
работ вида 22	3063
работ вида 23	2998
Выполнено заявок типа 3	12230
работ вида 31	6078
работ вида 32	3074
работ вида 33	3078
Выполнено заявок типа 4	12128
работ вида 41	5954
работ вида 42	3113
работ вида 43	3061
Выполнено заявок всех типов	46300
Вероятность выполнения всех заявок	0,927
Коэффициенты использования специалистов 1	1
Коэффициенты использования специалистов 2	0,954
Коэффициенты использования специалистов 3	0,83
Коэффициенты использования специалистов 4	0,796
Коэффициент использования диспетчеров	0,998

ЛИТЕРАТУРА

1. Карпов Ю. Г. Имитационное моделирование систем. Введение в модели-рование с AnyLogic 5. СПб.: БХВ Петербург, 2006. 400 с.
2. AnyLogic User's Manual. XJ Technologies : [электрон. ресурс]. Режим дос-тупа : http://www.xjtek.com
3. AnyLogic Tutorial. XJ Technologies : [электрон. ресурс]. Режим доступа : http://www.xjtek.com
4. Многоподходное имитационное моделирование в AnyLogic. XJ Technologies : [электрон. ресурс]. Режим доступа : http://www.xjtek.ru
5. Иванова Е.В., Затонский А.В. Оценка и моделирование научно-исследвовательской работы студентов как многоагентной системы // Современные наукоемкие технологии. 2009. № 7. С. 75-78.
6. Затонский А.В. Выбор вида модели для прогнозирования развития экономических систем // Новый университет. Серия: Технические науки. 2012. № 1 (7). С. 37-41.
7. Затонский А.В. Теоретический подход к управлению социально-техническими системами // Программные продукты и системы. 2008. № 1. С. 29-32.

Поступила в редакцию 09.03.2016