

УДК 65.012.122

**АГЕНТНОЕ ИМИТАЦИОННОЕ МОДЕЛИРОВАНИЕ БИЗНЕС-ПРОЦЕССОВ  
В НОТАЦИИ eEPC**

**Е.А. Ланцев, М.Г. Доррер**

Предложен и разработан теоретико-множественный аппарат описания моделей бизнес-процессов микроэкономической системы в методологии ARIS eEPC, а также агентных имитационных моделей AnyLogic. Разработан метод трансляции модели в нотации eEPC в агентную имитационную модель AnyLogic. На основе разработанного метода проведен эксперимент по трансляции модели в нотации eEPC в агентную имитационную модель AnyLogic.

**Ключевые слова:** агентное имитационное моделирование, eEPC, ARIS, AnyLogic.

**Введение**

Для анализа бизнес-процессов могут применяться непосредственно программные продукты для имитационного моделирования – Arena, AnyLogic, GPSS и др. Но использование специализированных систем имитационного моделирования для анализа деятельности предприятий и организаций сопряжено со значительной сложностью представления и согласования разработанных моделей с экспертами предметной области – менеджерами, экономистами, технологами. Для них более традиционным является представление деятельности предприятия в виде графических моделей бизнес-процессов в формате IDEF0/IDEF3, BPMN, EPC.

Существующие разработки в области трансляции между формальным описанием бизнес-процессов и его имитационной моделью в основном нацелены на получение в конечном итоге дискретно-событийной имитационной модели. В качестве целевой имитационной модели может использоваться как модель в формате существующей имитационной системы (Arena, GPSS, CPN Tools), так и формат для системы имитационного моделирования собственной разработки. Агентные модели в этой области применяются мало, в качестве имитационной системы используются собственные разработки [1–3].

В отличие от существующих разработок, в настоящей работе реализован более общий подход, теоретически применимый и к другим событийным моделям бизнес-процессов.

### Методы и материалы исследования

ARIS eEPC – расширенная нотация описания цепочки процесса, управляемого событиями, которая предназначена для детального описания бизнес-процесса и отражает логику его выполнения [4]. Модель бизнес-процесса представляет собой граф, состоящий из вершин – объектов модели, а также дуг (направленная связь) и ребер (ненаправленная связь) – связей между ними, которые не только соединяют объекты модели, но и служат для определения отношения между ними.

Модель в нотации eEPC (extended Event Driven Process Chain) определим как [5]:

$$G = \{X, V\},$$

где  $X$  – непустое множество объектов модели (вершин графа);  $V$  – непустое множество дуг  $V_n$  и ребер  $V_r$ , таких, что  $V = V_n \cup V_r$ .

Объекты модели (вершины графа):

$$X = \{S, F, D, I, P, R\},$$

где  $S$  – множество объектов типа «событие»;  $F$  – множество объектов типа «функция»;  $D$  – множество объектов типа «документ» и «информационный носитель»;  $I$  – множество объектов типа «Должность», «Организационная единица»;  $P$  – множество объектов типа «Продукт/услуга»;  $R$  – множество объектов типа «правило» XOR, OR, AND, задающих логику ветвления/слияния бизнес-процесса.

Множество объектов типа «событие»  $S$  состоит из множеств  $S_s \cup S_f \cup S_m$ . Здесь  $S_s$  – непустое множество начальных событий, определяемое как  $S_s = \{s_i \mid s_i \in S: \neg \exists s_j \in S: j < i\}$ ;  $S_f$  – непустое множество конечных событий, определяемое как  $S_f = \{s_i \mid s_i \in S: \neg \exists s_j \in S: j < i\}$ ;  $S_m$  – множество событий внутри модели, определяемое как  $S_m = \{s_i \mid s_i \in S: \exists s_j \in S: j < i\}$ , где  $s_i$  –  $i$ -ое событие модели.

Множество правил ветвления  $R$  состоит из элементов множества  $R_{xor} \cup R_{or} \cup R_{and}$ , где  $R_{xor}$  – множество правил ветвления типа «исключающее ИЛИ» (XOR);  $R_{or}$  – множество правил ветвления типа «ИЛИ» (OR);  $R_{and}$  – множество правил ветвления типа «И» (AND).

Множество объектов типа  $I$  состоит из элементов  $I_{pos} \cup I_{org}$ , где  $I_{pos}$  – множество объектов типа «Должность»;  $I_{org}$  – множество объектов типа «Организационная единица».

Множество объектов типа  $D$  состоит из элементов  $D_{doc} \cup D_{inf}$ , где  $D_{doc}$  – множество объектов типа «Документ»,  $D_{inf}$  – множество объектов типа «Информационный объект».

Множество дуг  $V_n$  определяется как

$$V_n = \{V_{sf}, V_{fs}, V_{sr}, V_{rs}, V_{fr}, V_{rf}, V_{fp}, V_{pf}\},$$

где  $V_{sf} = \{(s_i, f_j) \mid s_i \in S, f_j \in F\}$  – множество дуг, имеющих направление от события к функции;  $V_{fs} = \{(f_i, s_j) \mid f_i \in F, s_j \in S\}$  – множество дуг, имеющих направление от функции к событию;  $V_{sr} = \{(s_i, r_j) \mid s_i \in S, r_j \in R\}$  – множество дуг, имеющих направление от события к объекту типа «правило ветвления»;  $V_{rs} = \{(r_i, s_j) \mid r_i \in R, s_j \in S\}$  – множество дуг, имеющих направление от объекта типа «правило ветвления» к событию;  $V_{fr} = \{(f_i, r_j) \mid f_i \in F, r_j \in R\}$  – множество дуг, имеющих направление от функции к объекту типа «правило ветвления»;  $V_{rf} = \{(r_i, f_j) \mid r_i \in R, f_j \in F\}$  – множество дуг, имеющих направление от объекта типа «правило ветвления» к функции;  $V_{fp} = \{(f_i, p_j) \mid f_i \in F, p_j \in P\}$  – множество дуг, имеющих направление от функции к продукту/услуге;  $V_{pf} = \{(p_i, f_j) \mid p_i \in P, f_j \in F\}$  – множество дуг, имеющих направление от продукта/услуги к функции.

Множество ребер  $V_r$  определяется как

$$V_r = \{V_{fi}, V_{fd}\},$$

где  $V_{fi} = \{(i, f_j) \mid i \in I, f_j \in F\}$  – множество ребер, соединяющих функции и должности;  $V_{fd} = \{(f_i, d_j) \mid f_i \in F, d_j \in D\}$  – множество ребер, соединяющих функции и документы.

Агентная модель в AnyLogic реализуется с помощью базового объекта – активного объекта [6]. В общем виде агентная имитационная модель AnyLogic, основанная на событийных моделях eEPC, состоит из активных объектов (агентов), имеющих каждый свои правила поведения (задаются стейтchartом – диаграммой состояний) и взаимодействующих с внешней средой и друг с другом посредством пересылки сообщений (специальный программный класс на языке Java). Определим агентную модель AnyLogic как

$$ABM = \{E, A\},$$

где  $E$  – множество объектов, составляющих окружение агента;  $A$  – множество агентов (активных объектов).

Множество объектов  $E$ , составляющих окружение агента, определим как

$$E = \{Tm, Var, Ds, Msg\},$$

где  $Tm$  – множество таймеров;  $Var$  – множество переменных;  $Ds$  – множество таблиц для сохранения статистических данных о работе модели и синхронизации;  $Msg$  – множество сообщений для взаимодействия агентов со средой и друг другом.

Множество агентов  $A$  определим как

$$A = \{Sch, Par, Var, Fn\},$$

где  $Sch$  – непустое множество стейтчартов;  $Par$  – множество параметров агента;  $Fn$  – множество функций (специальный элемент AnyLogic для записи функций на языке Java) агента.

Множество стейтчартов  $Sch$  определим как

$$Sch = \{Bc, St, Trn\},$$

где  $Bc$  – непустое множество указателей начального состояния;  $St$  – непустое множество состояний;  $Trn$  – множество переходов.

Перевод графических элементов модели в нотации eEPC в агентную имитационную модель AnyLogic производится согласно табл. 1.

Рассмотрим подробнее метод преобразования модели в нотации eEPC в агентную имитационную модель AnyLogic, в процессе преобразований будем использовать соответствия, приведенные в табл. 1. Агент реализуется с помощью базового объекта AnyLogic – активного объекта. В качестве прототипа для класса агента используются организационные единицы eEPC (должность, организационная единица и другие), т.е. исполнители – ресурсы, необходимые для выполнения бизнес-функции, но не изменяемые ею (например, людские ресурсы, оборудование, и т.п.). В имитационной модели создаются классы активных объектов, и далее в модели используются любое число экземпляров класса.

Коммуникация между агентами осуществляется с помощью особого класса – сообщения (объект «сообщение» реализован через класс на языке Java), с его помощью агенту передается информация о состоянии, в которое ему нужно перейти. Сообщения могут поступать из внешней среды (например, по сигналу таймера) или от других агентов. У каждого из агентов имеется своя очередь из сообщений, которая обрабатывается, когда агент освобождается (дисциплина очереди – FIFO). Начальные события модели eEPC в агентной модели моделируются с помощью таймеров, которые генерируют заявки на обслуживание по заданному расписанию. Также сами агенты генерируют сообщения друг для друга.

Поведение агента задается стейтчартом. Простые состояния стейтчарта соответствуют функциям eEPC. Стейтчарт агента состоит из одного гиперсостояния и нескольких простых состояний. В гиперсостоянии есть только одно состояние – ожидание, в котором агент находится в самом начале моделирования, а также переходит после каждого простого состояния. Из гиперсостояния в какое-либо из простых состояний стейтчарт переходит, следуя последовательности смены состояний, либо по сигналу из внешней среды, либо по сигналу от других агентов.

Вероятностная логика правил ветвления eEPC «ИЛИ» и «исключающего ИЛИ» (OR и XOR) реализуется в методе (под методом здесь подразумевается соответствующий термин объектно-ориентированного программирования) перехода «Action» на языке Java при смене состояний (простого состояния на гиперсостояние). Таким образом, моделируется принятие агентом решения – в какое следующее состояние он должен перейти далее или же он должен перейти в режим ожидания и отправить сообщение другому агенту для обработки.

Параллельное выполнение модели функций eEPC в имитационной модели также реализуется в методе перехода «Action» на языке Java при смене состояний рассылкой сообщений необходимым агентам. Синхронизация правила ветвления «И» (AND) происходит при помощи внутренних таблиц (DataSet), в которые записывается состояние выполненных функций или произошедших событий.

Использование материальных ресурсов (например, количество определенных материалов на складе) и информационных ресурсов (например, количество созданных/обработанных документов) в имитационной модели фиксируется в таблицах (массивах) или же в простых переменных (вещественные, целые) при помощи кода на языке Java.

Получившаяся в результате преобразований по предложенному методу агентная имитационная модель является «заготовкой» для дальнейшего более детального моделирования – сосредоточения непосредственно на агентах, их поведении и коммуникациях.

### Эксперимент по трансляции в агентную имитационную модель

Для примера трансляции по предложенному методу рассмотрим модель бизнес-процесса «Формирование заказа на поставку товара» в нотации eEPC (рис. 1).

Вначале строим модель в AnyLogic на основе предложенного метода. В представленной модели eEPC введены три организационные единицы: «Служба безопасности», «Отдел выписки» и «Диспетчер», соответственно они будут являться в имитационной модели агентами. Затем, после получения структуры агентной имитационной модели, определим численные характеристики объектов модели, необходимые

для проведения имитационного моделирования: показатели для функций eEPC (время выполнения, количество единиц используемых ресурсов); количество исполнителей (организационные единицы) и соответственно экземпляров агентов, вероятности наступления событий в правилах ветвления, в данном случае вероятности для «исключающего ИЛИ» (XOR).

Объект модели eEPC	Графическое обозначение	Соответствующий элемент AnyLogic	Описание элемента AnyLogic
Функция (Function) $F \in X \in G$		Состояние (State) $St \in Sch \in A \in ABM$ 	Простое состояние стейтчарта (диаграммы состояний).
Начальное событие (Event) $S_s \in S \in X \in G$		Таймер (Timer) $Tm \in E \in ABM$ 	Наступление начальных событий определяется таймером по заданному аналитиком расписанию.
Событие (Event) $S_m, S_f \in S \in X \in G$		Переход (Transition) $Trn \in Sch \in A \in ABM$ 	Переход из гиперсостояния стейтчарта в простое состояние.
<b>Материальные и информационные ресурсы</b>			
Документ, Информационный носитель (Document) $D \in X \in G$		Переменная (Variable) $Var \in E \in ABM$ 	Переменные используются для моделирования изменяющихся характеристик и для хранения результатов моделирования. Изменение количественных ресурсов происходит в состояниях стейтчарта, программируется на языке Java.
Продукт/ Услуга (Product/ Service) $P \in X \in G$			
<b>Исполнители – ресурсы, необходимые для выполнения бизнес-функции, но не изменяемые ею</b>			
Должность (Position) $I_{pos} \in I \in X \in G$		Класс активного объекта (Active Object) $A \in ABM$ 	Поведение активного объекта описывается с помощью стейтчарта. В имитационной модели создаются классы активных объектов и далее в модели используется любое число экземпляров класса.
Организационная единица (Organizational unit) $I_{org} \in I \in X \in G$			
<b>Правила ветвления</b>			
Правило ИЛИ (OR rule) $R_{or} \in R \in X \in G$		Переход (Transition) $Trn \in Sch \in A \in ABM$ 	При переходе из простого состояния стейтчарта в гиперсостояние, в методе перехода «Action» на языке Java программируется логика принятия агентом решения.
Правило исключающего ИЛИ (XOR rule) $R_{xor} \in R \in X \in G$			
Правило И (AND rule) $R_{and} \in R \in X \in G$			

Таблица 1. Описание соответствия объектов модели нотации eEPC и элементов агентной имитационной модели на языке AnyLogic

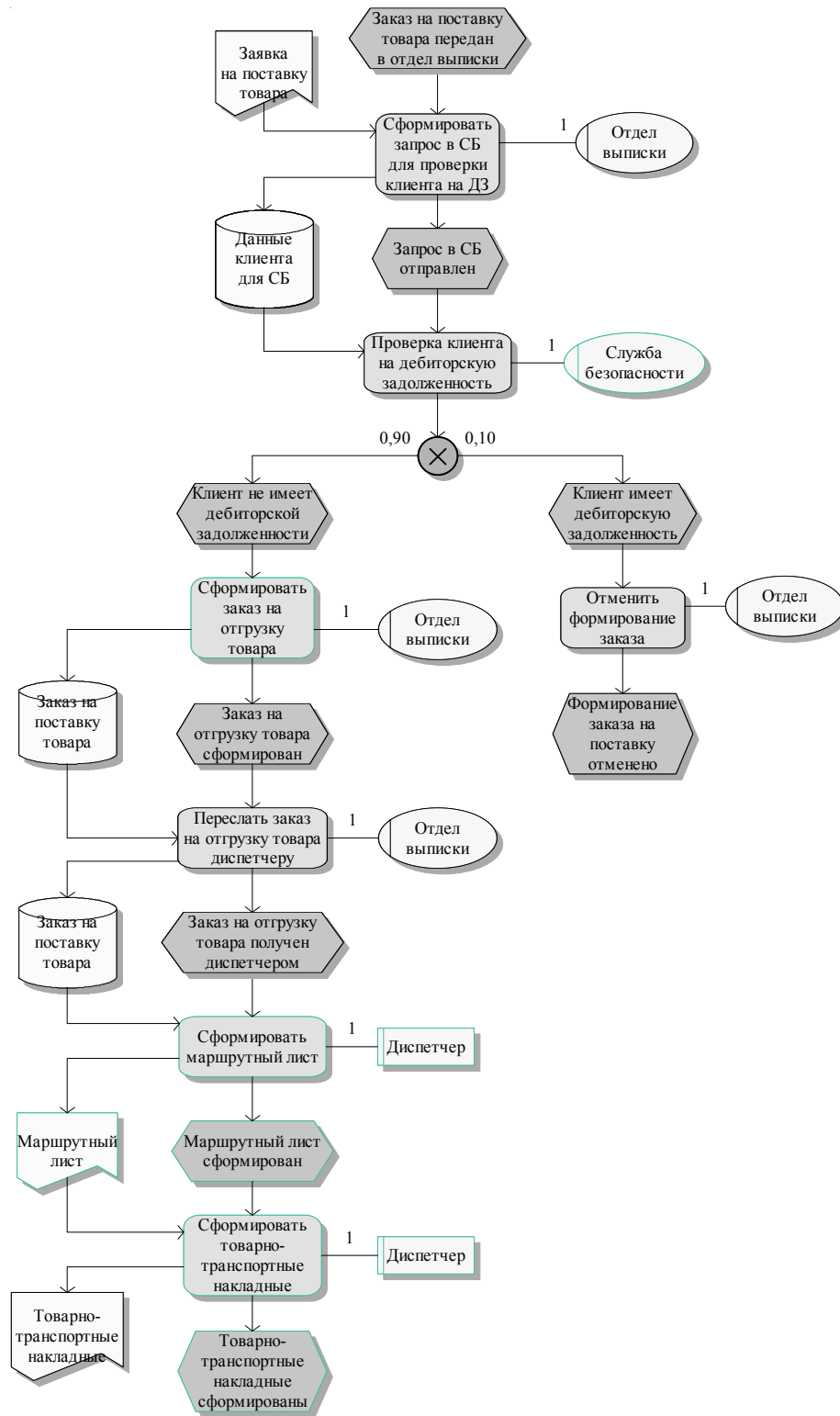


Рис. 1. Модель бизнес-процесса «Формирование заказа на поставку товара». СБ – служба безопасности, ДЗ – дебиторская задолженность

Фрагменты созданной агентной имитационной модели AnyLogic на основе модели eEPC представлены на рис. 2, 3. На рис. 2, а, показано дерево объектов модели, на рис. 2, б, отображается состав объектов, вложенных в класс «Main». Среда (окружение) моделируется классом «Main» (переменные, таймер, таблицы), в который вложены активные объекты – агенты (sB\_depart, v\_depart, dispatcher). Поскольку активные объекты технически являются классами на языке Java, то не представляется возможным давать им имена на русском языке, и для пояснений приходится ограничиваться текстовыми комментариями на самой модели. В данном примере, чтобы не перегружать модель комментариями, не были

сделаны комментарии к переменным (например, переменная «cancelOrders» хранит количество отмененных заказов, переменная «tr\_all» хранит количество сообщений, сгенерированных таймером, и т.д.). Далее на рис. 3 представлен пример создания агента «Диспетчер» и его стейтчарта на основе объектов модели eEPC (табл. 1).

Для верификации полученной агентной имитационной модели AnyLogic, проведено сравнение ее выходных данных с выходными данными имитационной модели встроенного в ARIS ToolSet имитатора ARIS Simulation. Сопоставив выходные данные экспериментов по загруженности исполнителей (табл. 2), а также по выполненным/отмененным заказам (табл. 3), полученные для одной и той же исходной модели eEPC, сделали вывод, что имитационная модель ARIS Simulation и агентная имитационная модель AnyLogic показывают сравнимые результаты моделирования при одинаковых исходных данных.

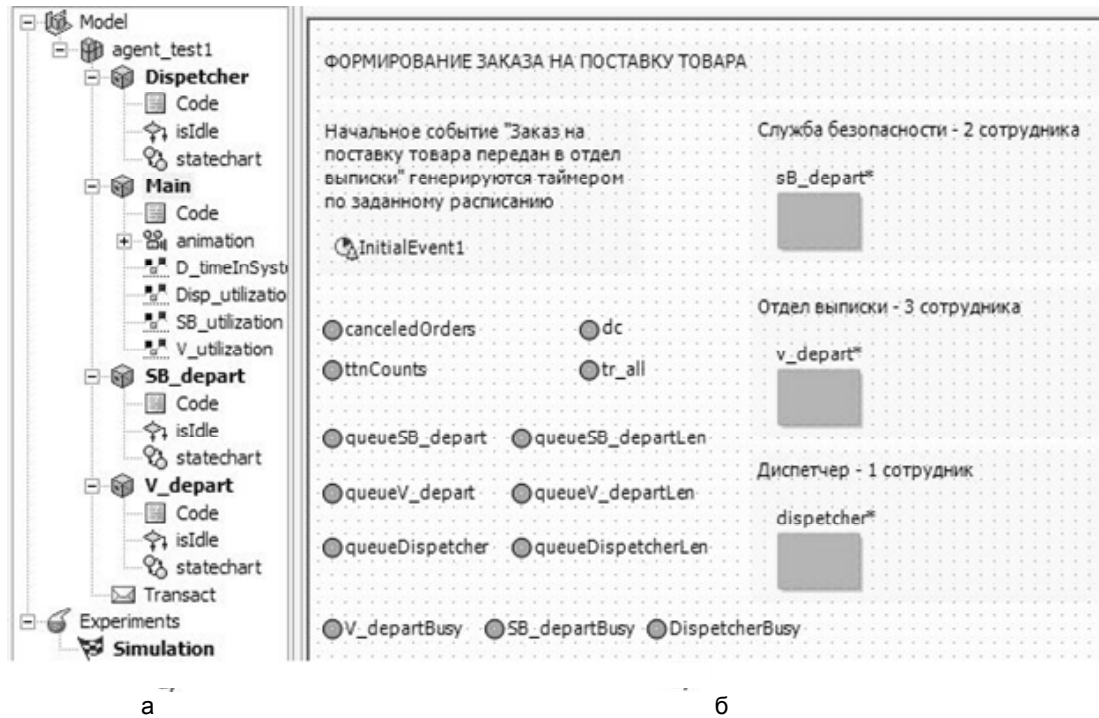


Рис. 2. Активные объекты (а) и их окружение (б) агентной имитационной модели AnyLogic

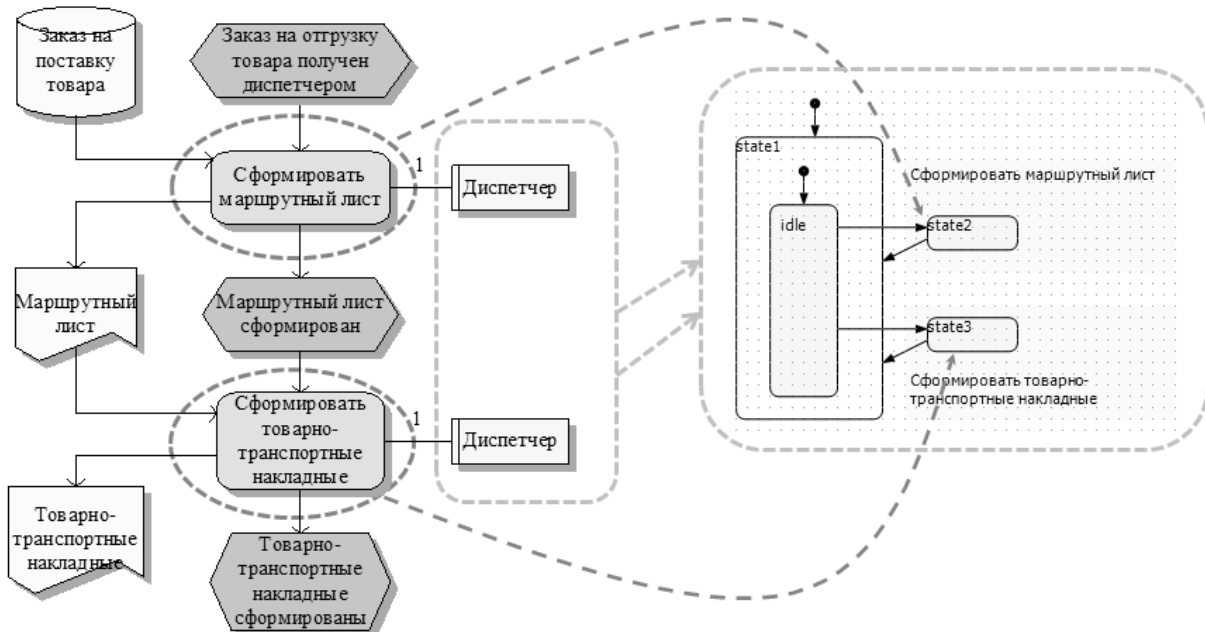


Рис. 3. Пример создания стейтчарта (диаграммы состояний) для программирования поведения агента «Диспетчер»

	ARIS Simulation, %	AnyLogic, %
Отдел выписки	97,90	98,40
Служба безопасности	88,30	83,80
Диспетчер	55,60	58,40

Таблица 2. Сводные усредненные данные по загруженности исполнителей, %

	ARIS Simulation	AnyLogic
Выполнено заказов	18	18
Отменено	1	3

Таблица 3. Сводные усредненные данные по выполненным/отмененным заказам

### Заключение

Установлена возможность взаимной связи элементов модели в нотации eEPC и агентной модели AnyLogic, разработан метод трансляции. Проведенный эксперимент по трансляции модели из нотации eEPC в агентную имитационную модель показал совпадение выходных данных по ключевым характеристикам исследуемых моделей в системах ARIS Simulation и AnyLogic. Дальнейшим развитием работы является автоматизация получения агентной имитационной модели на основе репозитория моделей в системе ARIS Toolset, что позволит еще больше сократить время проектирования имитационных моделей, а также упростит процесс их дальнейшей эксплуатации.

### Литература

1. Тихонов С.В. Моделирование бизнес-процессов с использованием аппарата теории систем массового обслуживания // Сборник трудов II конференции «II школа-семинар молодых ученых «Управление большими системами». – Т. 2. – Воронеж: Научная книга, 2007. – С. 5–59.
2. Валькман Ю.Р., Муцаковская К.В. Определение базиса для построения мультиагентной системы имитационного моделирования бизнес-процессов в нотации BPMN // Збірник наукових праць. – Інституту проблем моделювання в енергетиці. – 2012. – № 62. – С. 101–108.
3. Жевнерчук Д.В. Методика построения имитационных моделей с применением больших массивов данных на основе интеграции IDEF3, OLAP, GPSS технологий. – Ижевск, 2006. – 146 с.
4. Шеер А.В. Бизнес-процессы. Основные понятия. Теория. Методы. – М.: Весть-МетаТехнология, 1999. – 182 с.
5. Доррер М.Г., Ланцев Е.А., Шаргаева А.И. Событийный анализ бизнес процессов в нотации eEPC // Тр. XV Международной ЭМ конференции по эвентологической математике и смежным вопросам / СФУ, НИИППБ, КГТЭИ. – Красноярск, 2011. – С. 89–96.
6. Карпов Ю.Г. Имитационное моделирование систем. Введение в моделирование с AnyLogic 5. – СПб: БХВ-Петербург, 2006. – 400 с.

*Ланцев Евгений Алексеевич* – Сибирский государственный технологический университет, аспирант, red\_snow@list.ru  
*Доррер Михаил Георгиевич* – Сибирский государственный технологический университет, кандидат технических наук, доцент, mdorrer@mail.ru