

КОМБИНИРОВАННЫЙ ПОДХОД К ИМИТАЦИОННОМУ МОДЕЛИРОВАНИЮ БИЗНЕС-ПРОЦЕССОВ

© 2017 Е. А. Бабкин¹, Е. В. Копица²

¹*профессор кафедры программного обеспечения и администрирования
информационных систем, канд. техн. наук, доцент*

e-mail: eababkin@gmail.com

²*аспирант кафедры программного обеспечения и администрирования
информационных систем*

e-mail: kopitsa.egor@gmail.com

Курский государственный университет

В статье рассматривается подход к формализации концепции агента, предназначенного для функционирования в рамках подхода, основанного на комбинировании дискретно-событийной и агентной методологий. Выделены ключевые элементы понятия агента, представлены правила преобразования дискретно-событийной модели в агентную, а также приведен пример применения данных правил для процесса, описанного в нотации BPMN.

Ключевые слова: имитационное моделирование, комбинированная методология, агентное моделирование, формализация.

Ранее нами был рассмотрен подход к моделированию бизнес-процессов на основе комбинирования дискретно-событийной и агентной методологий, приведена формализация имитационной модели данного подхода [Бабкин, Копица 2016]. Требуется исследования вопрос формализации агента в рамках комбинированной системы имитационного моделирования.

Вопросу формализации концепции агента посвящены работы, в которых подробно описан набор возможных характеристик агента [Salamon 2009], рассмотрены его ключевые понятия и определения [Macal, North 2009]. Однако до сих пор вопрос формального определения агента остается открытым. Зачастую при построении систем агентного моделирования разработчики формализуют концепцию агента на основе собственного опыта и требований к разрабатываемой системе. Тем не менее большинство разработчиков систем агентного моделирования сходится во мнении, что агент должен обладать рядом характеристик, таких как:

- автономность – возможность функционировать независимо в пределах агентной среды, взаимодействуя с другими агентами лишь при необходимости;
- взаимодействие с другими агентами – возможность отправлять сигналы другим агентам и получать от них ответы;
- взаимодействие со средой, в которой он находится – возможность анализировать параметры среды и, в некоторых случаях, использовать ее для коммуникации с другими агентами.

В рамках разработки комбинированного подхода моделирования бизнес-процессов помимо описанных выше свойств будем характеризовать агента следующим набором признаков:

- уникальным именем;
- набором алгоритмов поведения;

- набором атрибутов, характеризующих агента;
- памятью;
- ресурсами, которыми обладает агент;
- возможностью взаимодействовать с другими агентами;
- возможностью взаимодействовать с агентной средой.

Этот список может быть расширен путем внедрения концепции целей агента, влияющих на механизм принятия решений, а также путем расширения поведения агента механизмами познания и адаптации. Данные элементы не обязательны для всех агентов, но могут быть полезны при решении определённого круга задач.

Таким образом, под агентом A будем понимать:

$$A = \langle N_a, B, \text{Attrib}, K, \text{Res}, P, \text{As} \rangle,$$

где N_a – имя агента;

B – набор алгоритмов поведения агента;

Attrib – набор атрибутов агента;

K – память (база знаний агента);

Res – набор ресурсов агента;

P – набор собственных действий;

As – набор ассоциаций агента.

Алгоритм поведения агента может быть далее представлен в форме

$$B_j = \langle N_b^j, \text{In}_j, \text{O}_j, \text{Asc}_j \rangle,$$

где N_b^j – имя алгоритма;

In_j – список входных параметров алгоритма;

O_j – список выходных данных;

Asc_j – сценарий выполнения.

В общем случае сценарий алгоритма поведения агента может быть представлен большим набором доступных средств, таких как блок схемы, событийные модели, BPMN-диаграммы [Onggo, Karpar 2011], набор диаграмм нотации UML [Bersini 2012] и др.

В данной работе на примере диаграммы нотации BPMN будет приведено соотношение между элементами описания агента и элементами нотации BPMN.

Будем рассматривать граф нотации BPMN:

$$G = \langle P, R \rangle,$$

где P – набор пулов;

R – набор ресурсов.

Под пулом будем подразумевать:

$$P_i = \langle N_p^i, S_i, D_i \rangle,$$

где N_p^i – имя пула;

S_i – набор дорожек;

D_i – набор данных пула.

Под дорожкой подразумевается:

$$S_i = \langle N_s^i, Act_i, Ru_i, E_{in}^i, E_{out}^i \rangle,$$

где N_s^i – имя дорожки;

Act_i – набор активностей дорожки;

Ru_i – набор правил перехода дорожки;

E_{in}^i – набор входящих событий дорожки;

E_{out}^i – набор исходящих событий дорожки.

В предлагаемом методе комбинирования нотаций предполагается, что описание бизнес-процесса происходит в процессном стиле, следовательно, имя пула N_p^i соответствует описываемым в них процессу, а имена дорожек N_s^i – исполнителям, в них участвующим. Таким образом, имена агентов N_a напрямую соответствуют именам дорожек N_s^i , используемым в BPMN-модели.

Формирование набора алгоритмов поведения агента B осуществляется на основе анализа дорожек нотации BPMN. В ходе формирования алгоритма поведения агента B_j имя алгоритма N_b^j формируется на основе имени пула N_p^i , в котором расположена дорожка и названия дорожки N_s^i . Набор активностей A_{se}_i и правил перехода Ru_i дорожки должны быть преобразованы в алгоритм поведения агента (сценарий) Asc_j . Нами рассмотрен один из возможных вариантов внутреннего описания структуры хранения данных агента [Бабкин, Копица 2016а].

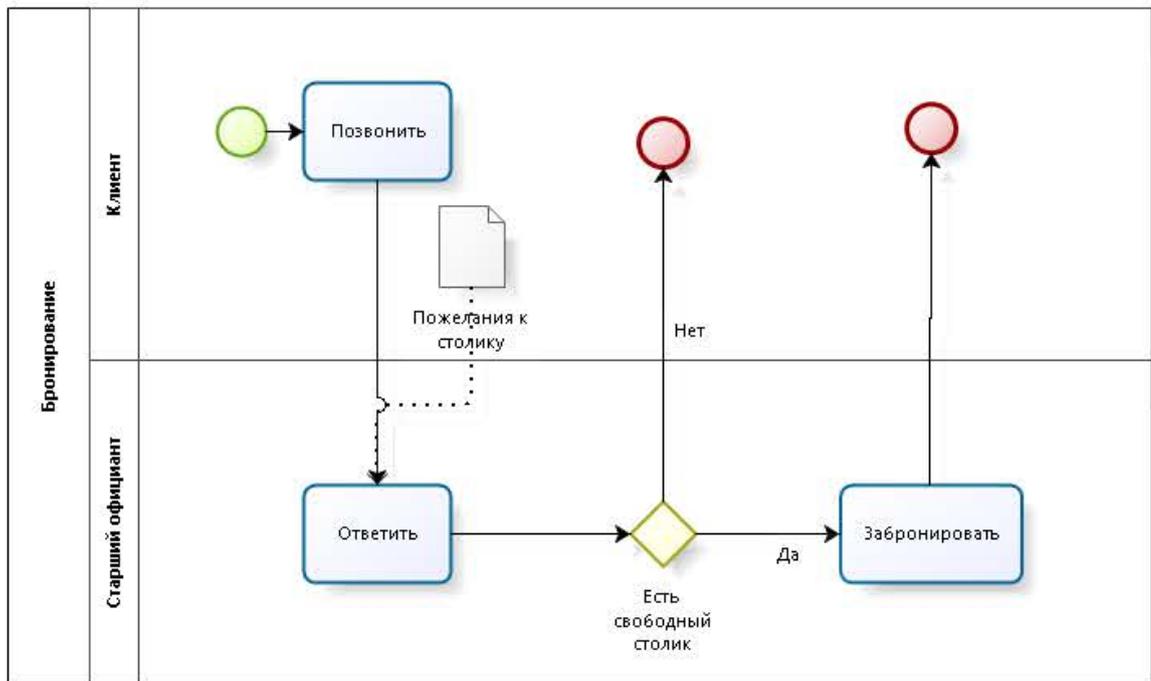
Активности дорожки должны быть преобразованы в собственные действия агента. Концепция собственных действий позволяет моделировать ограничения и характеристики активности, такие как время выполнения, резервирование ресурсов и др. В свою очередь, дорожки и подпроцессы должны быть преобразованы в доступные для внешнего вызова сценарии поведения агента Asc_j . Это позволит реализовать коммуникацию между агентами и их взаимодействие с внешней средой, а также будет способствовать расширению механизмов повторного использования и сокращению дублирования в разных частях модели.

События обеих групп E_{in}^i и E_{out}^i разделяются на события, содержащие сообщения, используемые для обмена сообщениями между дорожками, и пустые события, используемые для передачи потока управления другим дорожкам или инициации какой-либо активности взаимодействующего пула. Пустые события обоих типов не преобразуются непосредственно в атрибуты агента, но служат для формирования набора ассоциаций As с другими агентами. Каждое входящее событие дорожки, содержащее сообщение E_{in}^i , добавляется к списку входных параметров алгоритма поведения агента In_j . Исходящие события, содержащие сообщения E_{out}^i , преобразуются в выходные параметры алгоритма поведения агента O_j .

Набор данных пула D_i частично или полностью преобразуется в базу знаний K одного или нескольких агентов. В свою очередь, часть данных может быть преобразована во входные параметры действий агента.

В качестве примера рассмотрим преобразование процесса бронирования столика, приведенного в работе [Бабкин, Копица 2016];

Рассмотрим преобразование сценария бронирования столика по телефону.



Процесс обслуживания заявки на бронирование столика

Формальное описание сценария бронирования столика, представленного на рисунке:

$$P_{\text{Бронирование}} = \langle \text{Бронирование}, S_{\text{Бронирование}}, D_{\text{Бронирование}} \rangle,$$

$$S_{\text{Бронирование}} = \{S_{\text{ст.оф}}, S_{\text{клиент}}\},$$

$$D_{\text{Бронирование}} = \{\text{Пожелания к столику}\},$$

$$S_{\text{ст.оф}} = \langle \text{Старший официант}, Act_{\text{ст.оф}}, Ru_{\text{ст.оф}}, E_{in}^{\text{ст.оф}}, E_{out}^{\text{ст.оф}} \rangle,$$

$$Act_{\text{ст.оф}} = \{\text{Ответить}; \text{Забронировать}\},$$

$$Ru_{\text{ст.оф}} = (\text{Пожелания к столику})$$

$$\{$$

Ответить;

Если есть свободный столик, соответствующий пожеланиям, то

Забронировать;

Иначе

Завершить обслуживание;

$$\},$$

$$E_{in}^{\text{ст.оф}} = \{\emptyset\},$$

$$E_{out}^{\text{ст.оф}} = \{\emptyset\},$$

$$S_{\text{клиент}} = \langle \text{Клиент}, Act_{\text{клиент}}, Ru_{\text{клиент}}, E_{in}^{\text{клиент}}, E_{out}^{\text{клиент}} \rangle,$$

$$Act_{\text{ст.оф}} = \{\text{Позвонить}\},$$

$$Ru_{\text{клиент}} = \text{Клиент.Бронирование}()$$

$$\{$$

Позвонить;

Старший официант.Бронирование(пожелания к столику)

$$\},$$

$$E_{in}^{\text{клиент}} = \{\text{Начало обслуживания}\},$$

$E_{out}^{Клиент} = \{ \text{Удачное бронирование, Отказ в обслуживании} \}$.

Анализ формального описания сценария бронирования столика позволяет выделить двух агентов, взаимодействующих друг с другом: клиента и официанта. Следуя правилам преобразования, описанным в данной работе, получим следующие описания агентов:

$A_{Клиент} = \langle \text{Клиент, } \{ \text{Бронирование} \}, \{ \emptyset \}, \{ \emptyset \}, \{ \emptyset \}, \{ \text{Позвонить} \}, \{ A_{Официант} \} \rangle$

$V_{Клиент}^{\text{Бронирование}} = \langle \text{Бронирование, } \{ \emptyset \}, \{ \emptyset \}, A_{Клиент}^{\text{Бронирование}} \rangle$,

$A_{ст. оф.} = \langle \text{Старший официант, } \{ \text{Бронирование} \}, \{ \emptyset \}, \{ \emptyset \}, \{ \emptyset \}, \{ \text{Ответить, Внести запись в журнал бронирования} \}, \{ A_{Клиент} \} \rangle$,

$V_{ст. оф.}^{\text{Бронирование}} = \langle \text{Бронирование, } \{ \text{Пожелания к столику} \}, \{ \emptyset \}, A_{ст. оф.}^{\text{Бронирование}} \rangle$.

Стоит отметить, что сценарий $As_{Клиент}^{\text{Бронирование}}$, $As_{ст. оф.}^{\text{Бронирование}}$ формируется на основе $Act_{Клиент}$, $Ru_{Клиент}$ и $Act_{ст. оф.}$, $Ru_{ст. оф.}$ соответственно.

В ходе проведения анализа формального описания сценария бронирования столика и его преобразования в формальное описание агентов было выявлено большое количество неопределенных параметров: память агента, набор ресурсов агента и др. Стоит отметить, что использование таких агентов строго ограничено, так как они не обладают ни механизмом принятия решений, ни механизмом адаптации и изменения поведения. Также в ходе преобразования элементов нотации BPMN не было отражено формирование агентной среды и первоначальной настройки параметров агентов. Следовательно, использование только нотации BPMN недостаточно для формирования агентной модели, корректно отображающей предметную область. Возникает необходимость в формировании как минимум двухуровневой архитектуры модели, в которой первый уровень будет предназначен для бизнес-пользователей, заинтересованных в быстром формировании агентных моделей, а второй – для специалистов в области агентного моделирования, способных проводить точную настройку и корректировку внутренних алгоритмов поведения агентов. Кроме того, в целях сокращения издержек на переобучение персонала и облегчения адаптации сотрудников к новой системе моделирования она должна обладать определенной гибкостью в контексте используемых графических средств отображения бизнес-процессов, например, предоставлять возможность применять как диаграммы нотации BPMN, так и событийные графы или диаграммы деятельности нотации UML.

Библиографический список

Бабкин Е.А., Копица Е.В. О методологии имитационного моделирования бизнес-процессов на основе агентного и дискретно-событийного подходов // Auditorium. Электронный журнал Курского государственного университета. Курск, 2016. № 2(10). URL: <http://auditorium.kursksu.ru/pdf/010-012.pdf> (дата обращения: 03.05.2017).

Бабкин Е.А., Копица Е.В. Подход к реализации интерфейса программирования приложений при разработке системы мультиагентного моделирования // Сб. науч. тр. II Всерос. науч.-техн. конф. «Вопросы кибербезопасности, моделирования и обработки информации в современных социотехнических системах» (ИНФОРМ-2016). Курск: Курск. гос. ун-т, 2016а. С. 119–121.

Bersini H. UML for ABM // Journal of Artificial Societies and Social Simulation. Universite Libre de Bruxelles, Belgium, 2012. № 15 (1) 9. URL: <http://jasss.soc.surrey.ac.uk/15/1/9.html> (дата обращения: 03.05.2017).

Macal C.M., North M.J. Agent-based modeling and simulation // Winter simulation conference. 2009

Onggo B.S.S., Karpap O. Agent-based conceptual model representation using BPMN // Winter simulation conference. 2011

Salamon T. Design of agent-based models, Czech Republic, 2011. 209 P.