

## References

1. **Fihtengol'c G. M.** *Kurs differencial'nogo i integral'nogo ischislenija*. Vol. 1. Moscow, Fizmatlit, 2005. 650 p. (in Russian).
2. **Levakov A. A.** *Stohasticheskie differencial'nye uravnenija*. Minsk: BGU, 2009. 231 p. (in Russian).
3. **Puri M. L., Ralescu D. A.** Fuzzy Random Variables, *Journal of Mathematical Application*, 1986, no. 4, pp. 409–422.
4. **Kandel A.** Fuzzy Linear Systems, *Fuzzy Sets and Systems*, 1988, no. 96, pp. 201–209.
5. **Buckley J. J., Feuring J.** Fuzzy Differential Equations, *Fuzzy Sets and Systems*. 2000, no. 110, pp. 43–54.
6. **Mochalov I. A., Hrisat M. S., Shihab Eddin M. Ja.** Nechetkie differencial'nye uravnenija v zadachah upravlenija. Part 1, *Informacionnye tehnologii*, 2015, vol. 21, no. 3. pp. 171–178 (in Russian).
7. **Mochalov I. A., Hrisat M. S., Shihab Eddin M. Ja.** Nechetkie differencial'nye uravnenija v zadachah upravlenija. Part 2, *Informacionnye tehnologii*, 2015, vol. 21, no. 4. pp. 243–250 (in Russian).
8. **Gorban' I. I.**, *Fenomen statisticheskoj ustojchivosti*, Kiev, Naukova Dumka, 2014, 370 p. (in Russian).
9. **Alefel'd G., Hercberger Ju.**, *Vvedenie v interval'nye vychislenija*, Moscow, Mir, 1987, 360 p.
10. **Levin V. I.**, Interval'naja proizvodnaja i nachala nedeterministskogo differencial'nogo ischislenija, *Ontologija proektirovanija*, 2013, no. 4, pp. 72–84 (in Russian).
11. **Levin V. I.** Interval'no-differencial'noe ischislenie i nekotorye ego primenenija, *Informacionnye tehnologii*, 2014, no. 7, pp. 3–10 (in Russian).
12. **Levin V. I.** Differencial'noe ischislenie dlja interval'no-opredelennyh funkcij, *Jevristicheskie algoritmy i raspredelemnnye vychislenija*, 2015, vol. 2, no. 2, pp. 8–25 (in Russian).

УДК 004.94

**Д. В. Исаев**, канд. экон. наук, доц. кафедры бизнес-аналитики, disaev@hse.ru,  
Национальный исследовательский университет "Высшая школа экономики", г. Москва

## Имитационное моделирование сложных проектов

*Рассмотрены вопросы имитационного моделирования сложных проектов, к числу которых относятся проекты с неоднозначным исходом, возможностью повторного выполнения и несколькими возможными вариантами реализации. Предложенные модели и рекомендации могут быть использованы для оценки сроков реализации и последствий программ развития, в состав которых входят сложные проекты. Практическое применение моделей проиллюстрировано на примере системы дискретно-событийного моделирования Arena.*

**Ключевые слова:** программа развития, сложный проект, имитационное моделирование, дискретно-событийное моделирование, система Arena

### Введение

Одной из важных особенностей программ развития сложных организационно-технических и социально-экономических систем является наличие в их составе проектов, характеристики которых имеют вероятностный характер. Это относится, прежде всего, к срокам выполнения отдельных проектов и их финансовым показателям. Также следует принимать во внимание наличие "сложных" проектов, к которым относятся проекты с неоднозначным исходом, проекты с возможностью повторного выполнения и проекты с несколькими возможными вариантами реализации. Наличие вероятностных характеристик во многом объясняет то, что в практике проектного менеджмента фактические показатели реализации проектов и программ крайне редко совпадают с запланированными.

Наличие вероятностных характеристик проектов свидетельствует о целесообразности имитационного моделирования реализации проектов и программ развития. При этом можно считать, что состояние объекта моделирования (начало и завершение проектов, наступление программных событий, сопутствующие платежи, получение результатов)

изменяется в дискретные моменты времени. Поэтому в данном случае уместно применение имитационного моделирования дискретно-событийного типа, которое дает возможность описывать изменение состояния исследуемой системы в виде последовательности событий, наступающих в дискретные моменты времени [1].

Таким образом, задачи данного исследования заключаются в выявлении основных типов и характеристик сложных проектов, имеющих место в программах развития организационно-технических и социально-экономических систем, а также в разработке практических рекомендаций в области их имитационного моделирования.

В качестве иллюстрации возможностей практического применения моделей рассматривается их реализация в среде информационной системы *Arena* (разработка компании *Rockwell Automation*), функциональность которой достаточно подробно описана как в зарубежной [2–4], так и в отечественной [5, 6] литературе. Одной из основных областей практического применения системы являются различные системы массового обслуживания, в том числе входящие в состав производственных систем и вычислительных комплексов. Систему *Arena* также

можно применять в управлении проектами. Например, в работе [7] описано ее использование для проектов со случайной продолжительностью и наличием случайных временных лагов между проектами. Данная модель получила развитие в авторской работе [8], где, помимо самих проектов, рассматривается их влияние на показатели зрелости объекта управления и соответствующие финансовые показатели. Дальнейшее развитие модели связано с рассмотрением проектов, имеющих сложную конфигурацию и в наибольшей степени подверженных воздействию фактора неопределенности.

Рассмотренные модели также могут быть реализованы в других информационных средах, поддерживающих методы дискретно-событийного моделирования. В частности, примерами российских разработок в данной области могут служить система *AnyLogic*, поддерживающая различные типы имитационных моделей, включая дискретно-событийный [9], и система *Actor Pilgrim*, предназначенная для моделирования временной, пространственной и финансовой динамики экономических процессов [10]. В качестве зарубежного аналога отметим систему *GPSS World* (разработка компании *Minuteman Software*), которая представляет собой мощную среду имитационного моделирования с использованием языка *GPSS*, хотя и не поддерживает графическую визуализацию моделей [11].

### 1. Постановка задачи имитационного моделирования проектов и программ развития

Общая постановка задачи имитационного моделирования программ развития и входящих в их состав проектов выглядит следующим образом.

Программа развития представляет собой совокупность проектов, выполняемых в определенной последовательности. Продолжительность каждого из проектов обычно является случайной. Дополнительное усложнение задачи возникает в следующих случаях:

- если результаты реализации проекта не могут быть заранее определены однозначно (проект с неоднозначным исходом);
- если имеется возможность повторного выполнения проекта в случае недостижения поставленных целей (проект с возможностью повторного выполнения);
- если проект может быть выполнен несколькими разными способами, в зависимости от конкретной ситуации, складывающейся на момент его начала (проект с несколькими вариантами реализации).

Все проекты, обладающие перечисленными свойствами, будем называть сложными проектами. При этом допускается наличие проектов с определенным сочетанием перечисленных выше факторов сложности.

Поскольку проекты выполняются в определенной последовательности, каждый из них может стартовать только после завершения всех предшествующих проектов (исключение составляют стартовые проекты программы развития, не имеющие предшествующих). Завершение нескольких проектов может трактоваться как наступление программного события. Могут иметь место задержки (как правило, случайные) между завершением предшествующих проектов и возможным началом последующих, а также между завершением проектов и наступлением последующих программных событий.

Следствием завершения проектов и наступления программных событий является изменение показателей зрелости объекта управления (системы, на совершенствование которой направлена программа развития), а также финансовых показателей, связанных с реализацией программы развития. Влияние проектов и событий на показатели может сопровождаться временными задержками (как правило, случайными).

Результатом имитационного моделирования являются такие характеристики программы развития, как сроки ее реализации, а также динамика показателей зрелости объекта управления и финансовых показателей программы.

### 2. Практическое моделирование проектов

Рассмотрим моделирование реализации проектов с применением программного обеспечения *Arena* как одного из наиболее распространенных инструментов, который, наряду с развитой функциональностью, обеспечивает наглядную визуализацию создаваемых моделей.

В системе *Arena* объект моделирования представляется в виде ориентированного графа, в качестве узлов которого выступают графические модули. Эти модули относятся к разным типам, при этом каждый тип имеет свое содержательное наполнение и графическое обозначение (именно эти обозначения использованы в последующих иллюстрациях). Также имеются модули данных, которые служат для описания общих параметров модели, но не участвуют в ее визуализации.

Изменение состояния объекта моделирования происходит в результате перемещений между узлами графа индивидуальных динамических объектов, называемых транзактами. В отличие от систем массового обслуживания при моделировании проектов транзакты являются абстрактными и не имеют физической интерпретации.

Рассмотрим на конкретном примере (рис. 1) использование графических модулей системы *Arena* для моделирования проектов, отношений предшествования между ними, а также их последствий. При этом будем рассматривать только ту часть функциональности системы, которая является задействованной для решения сформулированной задачи.

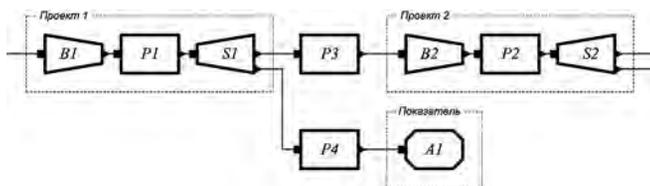


Рис. 1. Проекты, отношения предшествования и влияние на показатели

Основным при описании проекта является графический модуль типа **Process** (на рис. 1 модули этого типа имеют обозначения *P1* и *P2*). Поступление в этот модуль транзакта означает начало проекта, а выход транзакта из модуля — завершение проекта. В каждом из модулей данного типа указывается задержка — продолжительность реализации проекта. Период задержки может быть как детерминированным, так и случайным, в последнем случае задаются распределение случайной величины и его параметры.

При моделировании предполагается, что программа развития является стратегически важной для компании, и любой из проектов обеспечен достаточным объемом ресурсов — либо внутренних, либо привлекаемых со стороны. Поэтому конкретные ресурсы, как и уровень загрузки мощностей (актуальные для производственных систем), в данном случае не рассматриваем, вместо этого принимаем во внимание агрегированные финансовые показатели программы развития.

Поскольку проекты связаны между собой отношениями предшествования, каждый из них может иметь несколько предшествующих проектов и несколько последующих. Для описания таких отношений используют графические модули типов **Batch** и **Separate**.

Модуль **Batch** позволяет описать правило, согласно которому проект может стартовать только после завершения всех предшествующих проектов (на рис. 1 такие модули имеют обозначения *B1* и *B2*). В модуле осуществляется группировка входящих транзактов, при этом параметр объема группировки задается равным числу предшествующих проектов. В результате на входе модуля **Batch** возникает очередь из транзактов, ожидающих группировки. Сама группировка происходит в момент поступления на вход модуля последнего из ожидаемых транзактов, что означает завершение последнего из предшествующих проектов. Результатом группировки является единственный транзакт, который передается из модуля **Batch** в модуль **Process**. Если имеется лишь один предшествующий проект (или если проект является стартовым в программе развития), то модуль **Batch** можно не использовать.

Влияние рассматриваемого проекта на последующие описывается с помощью модуля типа **Separate** (на рис. 1 модули этого типа обозначены *S1* и *S2*).

На вход модуля поступает единственный транзакт (из модуля **Process**). В модуле **Separate** этот транзакт тиражируется, в результате чего возникают два новых одинаковых транзакта, один из которых условно называется оригиналом, а другой — копией. Таким образом, модуль **Separate** позволяет описать связь проекта с двумя последующими объектами. Если число последующих объектов больше двух, то используется "каскад" из нескольких модулей этого типа. Если же проект имеет лишь один последующий элемент (проект, событие или показатель), то модуль **Separate** не используется.

Для описания временных лагов, отделяющих завершение одних проектов от возможного начала других, используют такие же модули, как и те, которые применяют для описания продолжительности проектов, а именно — модули типа **Process**, с указанием детерминированных или случайных задержек. В рассматриваемом примере (см. рис. 1) проект 1 предшествует проекту 2, а разделяющий их временной лаг характеризует модуль *P3*.

По аналогии с проектами осуществляют моделирование программных событий, с той разницей, что период задержки, указываемый в модуле **Process**, задают равным нулю.

Для описания показателей зрелости объекта управления и финансовых показателей, связанных с реализацией программы развития, используют глобальные переменные (глобальными их называют потому, что они относятся не к какому-либо отдельному модулю, а к модели в целом). Глобальные переменные могут представлять собой не только отдельные переменные, но и массивы — одномерные или двумерные. В частности, для значений показателей зрелости используют одномерный массив с числом элементов, равным числу показателей. Аналогичный одномерный массив используют для финансовых показателей.

Начальные значения элементов массивов показателей задают в модуле данных **Variable** (этот модуль не является графическим и не используется при визуализации модели). В процессе моделирования изменения элементов массивов описываются с помощью графических модулей типа **Assign**. В рассматриваемом примере (см. рис. 1) проект 1 влияет на изменение показателя, а соответствующий модуль **Assign** имеет обозначение *A1*.

Степень влияния проектов и событий на показатели также описывается с помощью глобальных переменных, представляющих собой одномерные массивы. Значение каждого из элементов массива, описывающего воздействия на показатели зрелости, представляет собой гарантированное значение показателя, которое он получает в результате реализации соответствующего проекта или наступления соответствующего события. Аналогично, значение элемента массива, описывающего финансовые последствия, представляет собой исходящий денежный

поток, связанный с соответствующим проектом или событием. Значения элементов массивов воздействий также заданы в модуле данных **Variable**.

Как и в случае с предшествованием проектов, могут иметь место временные лаги между завершением проектов (или наступлением программных событий) и изменением показателей. Такие задержки также описывают с помощью модулей типа **Process** (на рис. 1 такой модуль имеет обозначение *P4*).

Наконец, рассмотрим элементы, задействованные при начале и завершении сеанса моделирования.

Начало сеанса моделирования описывается с помощью единственного графического модуля типа **Create**. В момент начала процесса моделирования этот модуль генерирует единственный транзакт, который подлежит тиражированию с использованием модулей **Separate** (одного или нескольких, в зависимости от объема тиражирования), с последующей доставкой возникших в результате тиражирования транзактов на вход стартовых проектов программы развития (т. е. проектов, не имеющих предшествующих). Возможные задержки между началом моделирования и возможными началами стартовых проектов описывают с помощью модулей **Process**.

Завершение сеанса моделирования описывается графическим модулем **Dispose**, который терминирует поступающие в него транзакты. Сеанс моделирования считается завершенным после терминирования последнего из транзактов, имеющихся в системе.

Приведенная модель позволяет описывать программы развития, включающие "простые" проекты, т. е. проекты, не допускающие вариативности получаемых результатов, возможности повторов, а также неопределенности в части способов их выполнения. Однако проекты с перечисленными свойствами ("сложные" проекты) также важны, поскольку довольно часто встречаются при развитии социально-экономических и организационно-технических систем. Поэтому рассмотрим основные типы сложных проектов, их роль в программах развития, а также возможности их моделирования с помощью системы *Arena*.

### 3. Проекты с неоднозначным исходом

Одной из разновидностей сложных проектов является проект с неоднозначным исходом. Необходимость рассмотрения таких проектов объясняется тем, что довольно часто при планировании проекта бывает трудно оценить не только его продолжительность, но и результаты, получаемые в ходе его выполнения. Результаты (исходы) проекта оказывают существенное влияние на его последствия — выполнение последующих проектов, наступление последующих проектных событий, изменение значений показателей зрелости объекта управления и финансовых показателей. Поэтому можно

считать, что каждому из возможных исходов проекта соответствует свой набор последствий.

Примерами проектов с неоднозначным исходом могут служить научно-исследовательские работы, в которых от результатов некоторых экспериментов зависит дальнейший ход исследования. Другим примером могут служить геолого-разведочные работы: здесь стратегия разведки также будет меняться в зависимости от результатов, полученных на предшествующих этапах.

При моделировании проектов с неоднозначным исходом может быть использован вероятностный подход. При этом считается, что каждый из возможных результатов реализации проекта достигается с некоторой вероятностью, сумма которых по всем возможным исходам равна единице.

Рассмотрим моделирование таких проектов в системе *Arena* на примере проекта с двумя возможными исходами (рис. 2).

Как и в случае простого проекта, модель сложного проекта с неоднозначным исходом имеет модуль **Process** с заданной продолжительностью выполнения проекта и модуль **Batch** с объемом группировки, равным числу предшествующих проектов (на рис. 2 эти модули имеют обозначения *P1* и *B1*). Однако в отличие от простого проекта, в данном случае предусматривается возможность получения одного из двух альтернативных результатов. Эта неопределенность описывается с помощью графического модуля **Decide**, который имеет обозначение *D1*.

По сути, модуль **Decide** характеризует ветвление процесса, при этом могут быть предусмотрены как две исходящие ветви, так и несколько исходящих ветвей (множественный выбор). В модели проекта число исходящих ветвей равно числу возможных исходов (в рассматриваемом на рис. 2 случае — двум). Выбор той или иной ветви для маршрутизации транзакта может осуществляться либо случайным образом, либо в соответствии с заданным условием. В случае вероятностного выбора (что предусмотрено постановкой рассматриваемой задачи) в качестве параметров модуля **Decide** задаются вероятности, в соответствии с которыми транзакт, поступивший в модуль, будет направлен по той или иной исходящей ветви (при этом сумма вероятностей по всем исходящим ветвям должна быть равна единице).

Отличие от простого проекта состоит также и в том, что в модели проекта с неоднозначным исходом предусматривается не один, а несколько модулей

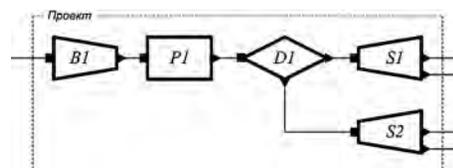


Рис. 2. Проект с неоднозначным исходом

**Separate.** Число таких модулей равно числу возможных исходов и, соответственно, числу выходов модуля **Decide** (в рассматриваемом примере с двумя возможными исходами имеются два модуля **Separate** с обозначениями *S1* и *S2*).

В модели модуль **Decide** расположен между модулем **Process** и модулями **Separate**. Транзакт, покидающий модуль **Process** при завершении проекта, поступает на вход модуля **Decide**. Здесь происходит его маршрутизация: в соответствии с заданными вероятностями транзакт покидает модуль **Decide** по той или иной исходящей ветви и, соответственно, поступает на вход того или иного модуля **Separate**. Дальнейшее поведение системы определяется дугами, связывающими модуль **Separate** (или каскад модулей данного типа, если число последующих элементов больше двух) с последующими модельными объектами — проектами, событиями и показателями.

#### 4. Проекты с возможностью повторного выполнения

Еще одним типом сложных проектов является проект с возможностью повторного выполнения. В этом случае проект считается полностью завершенным только тогда, когда его цели полностью достигнуты. Однако, если цели проекта не достигнуты, то есть возможность выполнить проект заново. Необходимость повторного выполнения проекта носит вероятностный характер: с одной вероятностью цели проекта достигаются и проект считается полностью завершенным, с другой вероятностью цели не достигаются, и поэтому требуется его повторное выполнение (сумма двух вероятностей равна единице).

Также могут возникать ситуации, когда повторное выполнение проекта возможно только по истечении некоторого промежутка времени (детерминированного или случайного) после завершения предыдущей итерации.

Проекты с возможностью повторного выполнения могут возникать при создании инновационных продуктов или внедрении принципиально новых (еще не апробированных) технологических решений. В этом случае успех проекта не гарантирован, и в случае неудачи проект инициируется заново.

Пример проекта с возможностью повторного выполнения приведен на рис. 3.

Как и для простого проекта, модель проекта с возможностью повторного выполнения предусматривает модуль **Process** с заданной продолжительностью

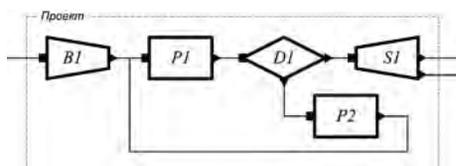


Рис. 3. Проект с возможностью повторного выполнения

стью выполнения проекта и модуль **Batch**, у которого параметр объема группировки равен числу предшествующих проектов (на рис. 3 эти модули обозначены *P1* и *B1*). Как и в случае проекта с неоднозначным исходом, присутствует графический модуль **Decide**, имеющий обозначение *D1*. Однако отличие состоит в том, что модуль **Decide** всегда имеет лишь две ветви, одна из которых характеризует достижение цели проекта и, соответственно, его окончательное завершение, а другая — необходимость повторного выполнения проекта. Транзакт, поступивший на вход модуля **Decide**, направляется по одной из двух исходящих ветвей. Выбор ветви для маршрутизации осуществляется в соответствии с заданными в модуле **Decide** вероятностями, сумма которых равна единице.

В отличие от проектов с неоднозначным исходом, для проекта с возможностью повторного выполнения предусмотрен лишь один модуль **Separate**. Транзакт поступает на вход этого модуля только в случае, когда цели проекта считаются достигнутыми и его повторное выполнение не требуется.

Как и в случае проекта с неоднозначным исходом, модуль **Decide** расположен между модулями **Process** и **Separate**. Один из выходов модуля **Decide** (соответствующий успешному выполнению проекта) соединяется со входом модуля **Separate**. Другой выход (характеризующий необходимость повторного выполнения) соединяется со входом модуля **Process**. В случае если имеет место временной лаг между завершением проекта и началом его повторного выполнения, используется дополнительный модуль **Process**, в котором описывается соответствующая задержка (на рис. 3 этот модуль имеет обозначение *P2*).

Транзакт, покидающий модуль **Process** при завершении очередной итерации проекта, поступает на вход модуля **Decide**, где происходит его маршрутизация. Если проект признается успешным, то транзакт перенаправляется на вход модуля **Separate**, а в случае необходимости повторного выполнения проекта — сначала в модуль **Process**, характеризующий задержку начала повторного выполнения, а затем — в модуль **Process**, описывающий непосредственную реализацию проекта.

#### 5. Проекты с несколькими вариантами реализации

К числу сложных также относятся проекты, допускающие несколько вариантов их реализации. Рассмотрение таких проектов можно считать целесообразным (а в некоторых случаях — даже необходимым), когда цели проекта могут быть достигнуты разными способами (например, с применением разных технологий или организационных схем), но выбор конкретного способа зависит от ситуации, имеющей место на момент начала проекта. На стадии формирования потенциальной программы развития можно определить совокуп-

ность возможных ситуаций и разработать для каждой из них свой вариант реализации проекта. Также могут быть сделаны предположения относительно вероятности появления каждой из ситуаций. Тогда конкретный способ реализации проекта будет определен в момент его начала с той или иной вероятностью в зависимости от складывающихся обстоятельств (сумма таких вероятностей по всем возможным ситуациям, связанным с рассматриваемым проектом, равна единице).

Примером проекта с несколькими вариантами реализации является внедрение информационной системы управления в случае, если применение того или иного программного обеспечения соответствующего класса будет определяться некоторыми объективными условиями (например, перспективами выхода компании-разработчика на российский рынок). В этом случае решение о выборе программного обеспечения (и соответственно, параметры проекта, характерные для этой альтернативы) носит условный характер, а окончательный выбор осуществляется при начале проекта, в зависимости от сложившейся на тот момент ситуации.

Пример проекта с несколькими возможными вариантами реализации приведен на рис. 4.

Как и в предыдущих случаях, в состав модели проекта с несколькими вариантами реализации входит единственный модуль **Batch**, с объемом группировки, равным числу предшествующих проектов (на рис. 4 этот модуль имеет обозначение *B1*). Также присутствует графический модуль **Decide**, у которого могут быть как две, так и несколько исходящих ветвей (в рассматриваемом примере этот модуль обозначен *D1*). Число исходящих ветвей модуля **Decide** равно числу вариантов реализации проекта. Выбор ветви для маршрутизации, соответствующей тому или иному варианту реализации проекта, проводится в соответствии с заданными вероятностями, сумма которых равна единице.

Транзакт, поступивший на вход модуля **Decide**, впоследствии направляется по одной из нескольких исходящих ветвей. При этом каждой из исходящих ветвей соответствует по одному модулю **Process** и одному модулю **Separate**, которые последовательно соединены друг с другом (в рассматриваемом примере для первой ветви эти модули имеют обозначения *P1* и *S1*, а для второй — *P2* и *S2*).

Для каждого из модулей **Process** указываются свои значения параметров, в частности продолжительность проекта, которая может быть как детер-

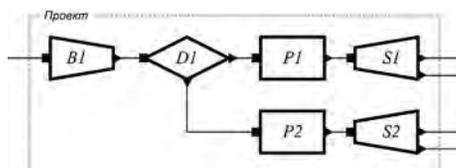


Рис. 4. Проект с несколькими вариантами реализации

минированной, так и случайной. Этим проект с несколькими вариантами реализации отличается от проекта с неоднозначным исходом и проекта с возможностью повторного выполнения, где реализация проекта осуществляется в соответствии с единственным набором параметров, заданных в модуле **Process**.

В модели проекта с несколькими вариантами реализации транзакт, покидающий модуль **Batch** в момент начала проекта, поступает на вход модуля **Decide**. В модуле **Decide** осуществляется маршрутизация, определяющая выбор конкретного варианта реализации проекта. В зависимости от сделанного выбора, транзакт передается в тот или иной модуль **Process**, характеризующий соответствующий вариант реализации проекта и имеющий свой набор параметров. Наконец, после модуля **Process** транзакт поступает на вход соответствующего модуля **Separate**, после чего дальнейшее поведение системы определяется исходящими дугами модуля **Separate** (или каскада модулей этого типа, если число последующих модельных объектов больше двух), а также самими последующими модельными объектами.

## Заключение

Многие из проектов, входящих в состав программ развития сложных организационно-технических и социально-экономических систем, находятся под значительным влиянием случайных факторов. Помимо вероятностного характера продолжительности выполнения проектов (что характерно для большинства из них), можно выделить такие факторы, как неоднозначность ожидаемых результатов, возможность повторного выполнения проекта, а также возможность реализации проекта несколькими разными способами. Перечисленные факторы могут сочетаться друг с другом, например, может рассматриваться проект с несколькими вариантами реализации, каждый из которых допускает неоднозначность получаемых результатов, в том числе приводящих к необходимости повторного выполнения проекта.

Перечисленные факторы существенно усложняют планирование проектов и программ развития, а также оценку последствий их реализации. Поэтому, принимая во внимание дискретный характер происходящих изменений, представляется уместным применение дискретно-событийного имитационного моделирования.

Одним из апробированных и хорошо зарекомендовавших себя инструментов дискретно-событийного моделирования является система *Arena*. Ее важными достоинствами являются возможность гибкого описания различных задержек (в том числе случайных), а также численных значений показателей, характеризующих ресурсоемкость и результативность проектов. Что касается сложных проектов (с неоднозначным исходом, возможностью повтор-

ного выполнения и наличием нескольких вариантов реализации), то значительную роль в их моделировании играют возможности вероятностной маршрутизации, которые в системе *Arena* обеспечивает специальный модуль выбора (модуль типа **Decide**). Таким образом, функциональность системы *Arena* является вполне достаточной для моделирования даже самых сложных проектов.

Практическая значимость предложенного подхода состоит в том, что на основе многократного прогона модели (возможно с разными наборами параметров) можно оценивать ключевые характеристики программ развития, включая сроки реализации проектов и наступления ключевых событий, а также показатели эффективности программ, определяемые на основе соотношения ожидаемых затрат и результатов. Это свидетельствует о применимости предложенной модели на стадии разработки и планирования программ развития, а при наличии нескольких альтернативных программ ее использования также и для принятия решения о выборе одной из них для реализации.

## Список литературы

1. **Wainer G. A.** Discrete-event modeling and simulation: A practitioner's approach. Boca Raton, FL: CRC Press, 2009. 520 p.
2. **Altiook T., Melamed B.** Simulation modeling and analysis with Arena. San Diego: Academic Press, 2007. 456 p.
3. **Kelton W. D., Sadowski R. P., Swets N. B.** Simulation with Arena. London: McGraw-Hill, 2014. 656 p.
4. **Rossetti M. D.** Simulation modeling and Arena. New York: John Wiley & Sons, 2015. 744 p.
5. **Замятина О. М.** Моделирование систем. Томск: ТПУ, 2009. 204 с.
6. **Щербakov С. М.** Имитационное моделирование экономических процессов в системе Arena. Ростов н/Д: РИНХ, 2012. 128 с.
7. **Cosgrove W. J.** Simplifying PERT network simulation with Arena // California Journal of Operations Management. 2008. Vol. 6, N. 1. P. 61—68.
8. **Исаев Д. В.** Анализ программ развития систем управления с применением программного обеспечения Arena // Аудит и финансовый анализ. 2015. № 6. С. 442—447.
9. **Бунцев И. А.** Создание и реализация имитационных моделей в программной среде AnyLogic. М.: Горячая линия — Телеком, 2015. 153 с.
10. **Емельянов А. А., Емельянова Н. З.** Имитационное моделирование и компьютерный анализ экономических процессов. Смоленск: Универсум, 2013. 266 с.
11. **Бразник А. Н.** Имитационное моделирование: возможности GPSS WORLD. СПб.: Реноме, 2006. 439 с.

**D. V. Isaev**, Associate Professor, e-mail: disaev@hse.ru,

Department of Business Analytics National Research University Higher School of Economics

## Simulation of Complicated Projects

*The paper focuses on simulation of complicated projects, associated with developing of socio-economic, organizational and technical systems. Few types of complicated projects are determined: projects with few possible outcomes, repeatable projects, and projects with few possible ways of implementation. Combined cases with certain combination of the factors of complexity may also be considered.*

*The purpose of the research is to determine key characteristics of complicated projects, and to develop recommendations in the field of their discrete-event simulation.*

*For illustration of practical application of the models, Arena software, as a popular and approved tool of discrete-event simulation, is applied. A set of models for different types of complicated projects are presented. Each of the models include detailed description of appropriate Arena modules, their functions in the model and their relevant parameters. The models and related recommendations advanced in the paper are applicable for assessment of implementation terms and consequences of development programs with complicated projects.*

**Keywords:** development program, complicated project, simulation, discrete-event modeling, Arena software

### References

1. **Wainer G. A.** Discrete-event modeling and simulation: A practitioner's approach, Boca Raton, FL, CRC Press, 2009, 520 p.
2. **Altiook T., Melamed B.** Simulation modeling and analysis with Arena, San Diego, Academic Press, 2007, 456 p.
3. **Kelton W. D., Sadowski R. P., Swets N. B.** Simulation with Arena, London, McGraw-Hill, 2014, 656 p.
4. **Rossetti M. D.** Simulation modeling and Arena, New York: John Wiley & Sons, 2015, 744 p.
5. **Zamyatina O. M.** Modelirovanie sistem [Modeling of systems], Tomsk, Tomsk Polytechnic University, 2009, 204 p.
6. **Shcherbakov S. M.** Imitatsionnoe modelirovanie ekonomicheskikh protsessov v sisteme Arena [Simulation of economic processes in Arena system], Rostov-on-Don, Rostov State University of Economics, 2012, 128 p.
7. **Cosgrove W. J.** Simplifying PERT network simulation with Arena, California Journal of Operations Management, 2008, vol. 6, no. 1, pp. 61—68.
8. **Isaev D. V.** Analiz programm razvitiya sistem upravleniya s primeneniem programmogo obespecheniya Arena [Analysis of management systems development programs using Arena software], Audit and Financial Analysis, 2015, no. 6, pp. 442—447.
9. **Buntsev I. A.** Sozdanie i realizatsiya imitatsionnykh modelei v programnoi srede AnyLogic [Development and implementation of simulation models in AnyLogic program environment], Moscow, Goryachaya Liniya — Telekom, 2015, 153 p.
10. **Emelyanov A. A., Emelyanova N. Z.** Imitatsionnoe modelirovanie i kompyuternyi analiz ekonomicheskikh protsessov [Simulation modeling and computer analysis of economic processes], Smolensk, Universum, 2013, 266 p.
11. **Brazhnik A. N.** Imitatsionnoe modelirovanie: vozmozhnosti GPSS WORLD [Simulation modeling: capability of GPSS WORLD], Saint Petersburg, Renome, 2006, 439 p.