

# Цифровые двойники и системы дискретно-событийного моделирования

Д.Е. Намиот, О.Н. Покусаев, В.П.Куприяновский, М.Г.Жабицкий

**Аннотация**— В этой статье речь идет об средствах разработки для цифровых двойников. Цифровые двойники привлекают к себе все большее внимание. Вместе с тем, нельзя не отметить тот факт, что большинство публикаций рассматривают именно применение цифровых двойников. То, каким образом эти двойники были построены (разработаны) остается вне рассмотрения. Между тем, очевидно, что именно вопросы разработки (создания) двойников являются первичными. Любой цифровой двойник включает в себя модель реального объекта, представленную в той или иной форме. Это то, что называется цифровой репликой. В работе рассматривается связь систем разработки дискретных имитационных моделей и цифровых двойников. Дискретно-событийное моделирование является одним из старейших и широко используемых подходов к построению систем моделирования. В работе рассматриваются дополнения для систем дискретно-событийного моделирования, которые позволяют использовать такие инструментальные средства для разработки цифровых двойников. Такого рода системы будут важны также и для университетов в плане обучения разработке цифровых двойников.

**Ключевые слова**—цифровой двойник, моделирование.

## I. ВВЕДЕНИЕ

В статье представлен расширенный и дополненный вариант доклада на конференции MACISE 2021.

Дискретное имитационное моделирование (в данной работе – это и дискретно-событийное моделирование), в англоязычной литературе - discrete-event simulation (DES) - это один из самых старых и широко используемых подходов к имитационному моделированию. В дискретно-событийном подходе модель системы (описание ее функционирования) представляется как последовательность событий во времени. Каждое событие (или события в многопоточных системах) происходит в определенный момент времени и фиксирует собой некоторое изменение состояния системы [1].

Состояния системы определяются значениями переменных, которые являются составной частью

модели. Инициализация (изменение значений) переменных происходит в соответствии с логическими правилами, которые определяют реакцию на событие (или события). Соответственно, такие логические правила также являются частью системы. Другой частью любой подобной системы является синхронизатор – механизм (алгоритм) синхронизирующий изменения состояний (реализующий “возникновение” событий)

Построенные модели бывают однопоточные, то есть имеющие только одно событие, или многопоточные, которые могут иметь несколько текущих событий.

Сами события бывают мгновенные и интервальные. По характеру возникновения событий дискретно-событийные модели делятся на детерминированные и стохастические, в зависимости от того, каким образом генерируются события [2].

Условием завершения работы модели могут выступать:

- Возникновение заданного события
- Достижение определенного состояния
- Истечение заданного времени по часам системы моделирования
- Прохождение заданного числа циклов по часам системы моделирования

Результатом моделирования являются статистические характеристики переменных системы, накопленные (измеренные) в процессе работы модели.

Это один из самых старых (долго используемых) подходов к моделированию систем, накопивший за годы использования обширную практику и, главное, приведший к появлению популярных инструментальных средств (библиотек, фреймворков, проблемно-ориентированных языков программирования).

Цифровой двойник, согласно классическому определению - это цифровая копия живого или искусственного физического объекта [3]. Термин цифровой двойник относится к цифровой копии потенциальных и реальных физических активов (физический двойник), процессов, людей, мест, систем и устройств, которые могут использоваться для различных целей. Цифровые двойники призваны облегчать средства контроля, понимания и оптимизации функций всех физические активов, обеспечивая беспрепятственную передачу данных между физическим и виртуальным миром [4].

Статья получена 12 января 2021.

Д.Е. Намиот - МГУ имени М.В. Ломоносова; РУТ (МИИТ) (e-mail: dnamiot@gmail.com)

О.Н. Покусаев – РУТ (МИИТ); РОАТ (e-mail: o.pokusaev@rut.digital).

В.П.Куприяновский - РУТ (МИИТ) (email: vrpupriyanovsky@gmail.com)

М.Г.Жабицкий - Высшая инженеринговая школа НИЯУ МИФИ (email: jabitsky@mail.ru)

Соответственно, цифровой двойник по определению включает в себя имитационную модель. Отсюда следует естественное предположение, что средства для разработки имитационных моделей могут быть использованы и при создании цифровых двойников. Поскольку большинство таких систем создавалось в “двойниковую” эру, у них нет всего, что требуется для цифровых двойников. Поэтому корректным является вопрос о том, что необходимо (возможно) добавить к инструментарию для создания дискретно-событийных моделей, чтобы его можно было использовать для цифровых двойников.

Технология цифровых двойников играет решающую роль в Индустрии 4.0, обеспечивая единое представление реального и виртуального миров.

Оставшаяся часть статьи структурирована следующим образом. В разделе II приводится сравнение дискретно-событийных моделей и цифровых двойников, в которых цифровая реплика может быть представлена в виде набора состояний. В разделе III обсуждаются расширения для программных систем построения дискретно-событийных моделей.

## II. ДИСКРЕТНО-СОБЫТИЙНЫЕ МОДЕЛИ И ЦИФРОВЫЕ РЕПЛИКИ

Большинство публикаций, посвященных цифровым двойникам (типичные примеры – работы [5], [6]) посвящены именно эксплуатации. Тема же данной статьи – именно разработка. Как было построено то, эксплуатация чего красиво описывается в большинстве статей.

Виртуальные модели в цифровом двойнике должны быть точными копиями физических объектов. Это означает, вообще говоря, точное воспроизведение физической геометрии, свойств, поведения и правил функционирования [7]. Например, трехмерные геометрические модели могут описывать физический объект с точки зрения его формы, размера, и т.д. На основании физических характеристик (свойств), например, скорости и силы, физическая модель должна отражать такие явления, как деформация, разрушение и т.п. Здесь необходимо отметить, что дискретно-событийные модели обычно не очень подходят для описания именно физических моделей в указанном выше смысле. Это не есть их недостаток в части использования для проектирования цифровых двойников, это просто ограничение в части применимости. Для физических моделей просто есть другие средства разработки.

Модели поведения, в соответствии с названием, описывают поведение и механизмы реагирования на изменения во внешней среде. Описание поведения в форме переходов между состояниями – это как раз то, что является основой дискретно-событийного моделирования.

Правила тесно связаны с моделями поведения и моделируют принятие решений, выполнение оценок и

т.п. Правила могут создаваться на основе исторических данных (если таковые есть и доступны) или задаваться экспертами. Эта компонента соответствует логическому модулю в дискретно-событийном моделировании.

Следующий элемент, являющийся важнейшим для цифровых двойников – это, конечно, данные. Все цифровые двойники, потенциально, имеют дело с разнородными и многомерными данными, поступающими из разных источников. В системах имитационного моделирования чаще всего принимаются упрощенные модели данных. Например, для множества моделей дискретно-событийного моделирования типичным является моделирование наступления событий в соответствии с некоторым статистическим распределением. Для цифровых двойников упрощения невозможны (иначе это будет не цифровой двойник). Данные модель может получать непосредственно от физических объектов, при этом данные могут быть как статическими, так и динамическими, данные могут порождаться уже самой моделью, либо поступать от различных сервисов, обрабатывающих как данные измерений, так и внешние (по отношению к физическому объекту) данные. Отметим также, что данные могут возникать и в результате слияния (соединения) измерений (data fusion).

Естественно, что в таком случае эти данные не поступают сами по себе, и для цифрового двойника должна быть характерна поддержка множества соединений. Это необходимо для получения данных от физического объекта, а также от внешних сервисов. Эти связи представлены на рисунке 1.

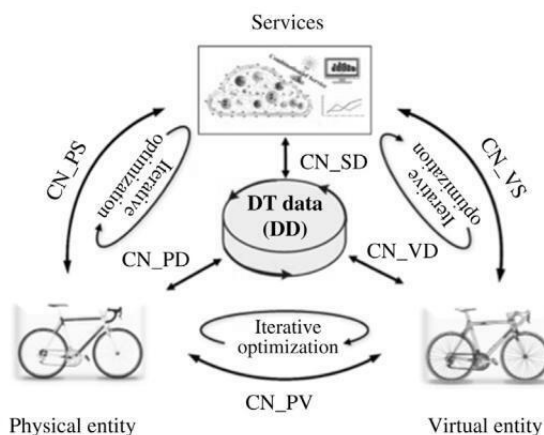


Рис. 1. Соединения в цифровом двойнике [8]

Основное отличие цифрового двойника от имитационной модели может быть сформулировано следующим образом. Моделирование – это анализ “что будет если ...”, тогда как цифровой двойник – это “что происходит сейчас и что будет если ...”. Ключевые отличия также включают следующие моменты.

Цифровой двойник – это моделирование в реальном времени. Традиционное моделирование выполняется в

виртуальных средах, которые могут представлять физические среды, но не объединяют данные в реальном времени. Регулярная (постоянная) передача информации между цифровым двойником и соответствующей физической средой делает возможным моделирование в реальном времени. Это повышает точность прогнозных аналитических моделей, а также качество управления и мониторинга.

Большее количество переменных. Сбор всех данных позволяет расширить количество возможных переменных, доступных для модели. Естественно, что такой анализ требует поддержки в описаниях моделей поведения или в правилах модели, но он, по крайней мере, становится принципиально возможным. Соответственно, принципиально возможным становится и расширенное использование результатов моделирования. Иными словами, цифровой двойник должен (может) получать больше результатов, чем просто имитационная модель.

Аккумуляция больших наборов данных в цифровом двойнике повышает его значимость по сравнению с имитационной моделью. Цифровой двойник становится неотъемлемой частью процесса эксплуатации физического объекта, который используется для прогнозной аналитики, технического обслуживания и т.п. В итоге цифровой двойник влияет на физический объект, например, предоставляет данные для его улучшения, оптимизации процессов и т.п. В качестве примеров можно привести, например, цифровые двойники автомобилей Tesla. У каждого автомобиля Tesla, работающего сегодня, есть цифровой двойник, который собирает большие наборы данных, производимые каждым автомобилем. Собранные данные используются для оптимизации дизайна, прогнозной аналитики, совершенствования технического обслуживания и т.д. [9]. Другой пример – цифровые двойники в BIM, используемые при эксплуатации здания и сооружений [6, 10]. Чисто имитационные модели, как правило, используются только на этапе проектирования.

### III. ИНСТРУМЕНТЫ ДЛЯ ДИСКРЕТНО-СОБЫТИЙНОГО ПОДХОДА И ЦИФРОВЫЕ ДВОЙНИКИ

В этом разделе мы хотели бы остановиться на средствах разработки для дискретно-событийных моделей. Естественно, что мы исходим из того, что для нашей системы дискретно-событийный подход будет подходящим для представления цифровой реплики. В каких-то случаях это будет не так, и будет необходимо использовать другие инструменты. Например, в работе [11] описываются также дискретные, но динамические системы и т.д.

Моделирование дискретных событий (DES), как было указано выше – это форма компьютерного моделирования системы с помощью дискретной

последовательности событий. Термин «дискретный» описывает тот факт, что события развиваются вперед по времени через определяемые в модели интервалы. Собственно моделирование предназначено для отслеживанию эволюции системы во времени. Это достигается с помощью так называемых системных (имитационных) часов, которые меняются в зависимости от возникновения событий.

В классической работе [12] схемы использования имитационных часов (модельного времени) подразделяются на два типа:

1) Определение времени следующего события (NETA - Next Event Time Advances). NETA – это механизм, определяющий время наступления будущих событий на основе некоторого списка. При этом типичном подходе модельное время устанавливается в нулевое значение, а далее обновляется по мере наступления событий. При этом между последовательными событиями ни при каких обстоятельствах не должно происходить никаких изменений в системе. Все изменения состояний привязаны только к событиям.

2) Увеличение времени с помощью некоторого фиксированного приращения (FITA - Fixed Increment Time Advances). Этот альтернативный подход просто продвигает системные часы на некоторое значение  $\Delta t$ . После изменения системного времени определяются события, которые произошли за это время (время, прошедшее после последнего приращения). Согласно наступившим событиям обновляются внутренние переменные (состояния). После чего время снова продвигается. Как отмечается в литературе, в целом, NETA более широко используется в моделировании, чем FITA, из-за своей меньшей сложности [13].

Области применения дискретно-событийного подхода довольно широкие. Это происходит в силу простоты и достаточно высокой эффективности описания систем в виде набора состояний. Классические применения – это все, что может быть представлено как система массового обслуживания: сервисные предприятия, транспортные системы, логистика и складские операции.

Если мы переходим от дискретно-событийной модели к цифровому двойнику, цифровая реплика которого построена на базе дискретно-событийного подхода, то, очевидно, что модельное время, в силу требования передачи данных в реальном времени должно соответствовать реальному времени. Таким образом, например, нулевой отсчет – это просто время начала работы цифрового двойника. Время следующего события – это реальное время наступления события, приращение времени – это новая отметка, в которой проверяются наступившие события после последней проверки.

Что и как должно быть изменено в средствах разработки, можно представить на примере языка Simula [14]. Этот язык одним из первых предложил

практическую реализацию системы дискретно-событийного моделирования, основанную на корутинах [15]. Предложенные здесь механизмы и являются по сей день основой для инструментов дискретно-событийного моделирования.

Ниже приведен классический пример простой системы массового обслуживания, взятый непосредственно из работы [16]. В этой программе моделируется работа некоторой примерочной (для простоты – одной единственной), которой в эксклюзивном порядке пользуются посетители (*Person*). Посетитель проводит какое-то время в магазине (*Hold (Normal (12, 4, u))* – случайное значение с нормальным распределением), запрашивает использование примерочной, ждет в очереди (*Wait.door*), если примерочная занята другим, далее проводит какое-то время в примерочной (*Normal (3, 1, u)* – случайное значение с нормальным распределением), после чего покидает систему (*fittingroom1.leave*), продвигая следующего в очереди. Классическая модель конкуренции за критический (единственный) ресурс.

```
Simulation Begin
  Class FittingRoom; Begin
    Ref (Head) door;
    Boolean inUse;
    Procedure request; Begin
      If inUse Then Begin
        Wait (door);
        door.First.Out;
      End;
      inUse:= True;
    End;
    Procedure leave; Begin
      inUse:= False;
      Activate door.First;
    End;
    door:- New Head;
  End;

  Procedure report (message); Text
  message; Begin
    OutFix (Time, 2, 0); OutText (": " &
  message); OutImage;
  End;

  Process Class Person (pname); Text
  pname; Begin
    While True Do Begin
      Hold (Normal (12, 4, u));
      report (pname & " is requesting
  the fitting room");
      fittingroom1.request;
      report (pname & " has entered the
  fitting room");
      Hold (Normal (3, 1, u));
      fittingroom1.leave;
      report (pname & " has left the
  fitting room");
    End;
  End;

  Integer u;
  Ref (FittingRoom) fittingRoom1;

  fittingRoom1:- New FittingRoom;
  Activate New Person ("Sam");
```

```
Activate New Person ("Sally");
Activate New Person ("Andy");
Hold (100);
```

End;

Отметим, что модель (программа) программа создает все объекты и активирует все объекты (*Person*), чтобы поместить их в очередь событий. Основная программа работает 100 минут в модельном времени (*Hold(100)*).

Жирным шрифтом и подчеркиванием в программе выделены фрагменты, которые подлежат замене в цифровом двойнике.

Во-первых, времени работы для цифрового двойника просто не будет. Соответственно, оператор *Hold(100)* должен быть просто удален.

Далее идет создание объектов - *New Person ("Sam")* и т.д. Очевидно, что создание объектов должно соответствовать реальному появлению пользователей. Тогда это не статический оператор, а процесс в рамках модели, который сканирует какие-то датчики реальной системы (открытие двери, видео-аналитика, выдача номера для примерочной и т.п.) и создает нового посетителя уже на основании информации от этих датчиков – одного или нескольких.

Ну и самое главное – это работа с модельным временем *Hold ()*. В данном случае модель следует схеме NETA и разыгрывает случайное время наступления следующего события – как долго пользователь присутствует в системе до того, как решит воспользоваться примерочной, и как долго он будет находиться в выбранной примерочной. Эти операторы (вызовы) также должны быть заменены процессами, которые опрашивают данные сенсоров, для определения наступления соответствующих событий.

Можно отметить, что сама схема цифровой реплики будет меняться в зависимости от того, что реально возможно измерять в конкретных условиях. Например, практически несложно организовать определение времени занятия примерочной (датчик присутствия, бэдж для входа, который сдается при выходе и т.п.). Но совсем не так просто организовать индивидуальный мониторинг нахождения в зале. Возможно, например, что в цифровом двойнике будет отмечаться приход посетителя, а вместо времени нахождения в зале будет измеряться конверсия – количество посещений примерочных в зависимости от числа посетителей. Иными словами, цифровая реплика (ее структура) будет зависеть от доступности данных физического объекта. При этом под доступностью нужно понимать как техническую возможность их получения, так и экономическую целесообразность их сбора.

Каким образом можно обеспечить доступ к данным измерений для дискретно-событийной модели? Как наиболее простой способ добиться этого видится введение некоторой шины (Kafka или подобные продукты), куда, с одной стороны, будут публиковаться данные измерений, а с другой стороны будут

подписчики (модель publish-subscribe), которые эти данные будут читать. Тогда, например, вместо розыгрыша случайной задержки при моделировании занятия примерочной будет получение по подписке сигнала о занятии и ожидание получения сигнала об освобождении. Разница по времени между этими двумя событиями (уже реальными событиями) и будет учитываться в статистике как время для конкретного посетителя.

[16] Simula <https://en.wikipedia.org/wiki/Simula> Retrieved: Jan, 2021

#### IV. ЗАКЛЮЧЕНИЕ

В этой статье кратко рассмотрены возможности использования инструментальных средств для дискретно-событийного моделирования при разработке цифровых двойников. Проведен сравнительный анализ классических дискретно-событийных моделей и цифровых реплик в двойниках, использующих дискретно-событийный подход. Представлен набор расширений для систем разработки дискретно-событийных моделей, необходимых для использования таких средств разработки при создании цифровых двойников.

#### БИБЛИОГРАФИЯ

- [1] Varga, András. "Discrete event simulation system." Proc. of the European Simulation Multiconference (ESM'2001). 2001.
- [2] Fishman, George S. Discrete-event simulation: modeling, programming, and analysis. Springer Science & Business Media, 2013.
- [3] El Saddik, Abdulmotaleb. "Digital twins: The convergence of multimedia technologies." IEEE multimedia 25.2 (2018): 87-92.
- [4] Khajavi, Siavash H., et al. "Digital twin: vision, benefits, boundaries, and creation for buildings." IEEE Access 7 (2019): 147406-147419.
- [5] Kurganova, Nadezhda, et al. "Digital twins' introduction as one of the major directions of industrial digitalization." International Journal of Open Information Technologies 7.5 (2019): 105-115.
- [6] Kupriyanovsky, Vasily, et al. "Digital twins based on the development of BIM technologies, related ontologies, 5G, IoT, and mixed reality for use in infrastructure projects and IFRABIM." International Journal of Open Information Technologies 8.3 (2020): 55-74.
- [7] Qi, Qinglin, et al. "Enabling technologies and tools for digital twin." Journal of Manufacturing Systems (2019).
- [8] Tao, Fei, et al. "Digital twin driven prognostics and health management for complex equipment." Cirp Annals 67.1 (2018): 169-172.
- [9] Tharma, Rajeeth, Roland Winter, and Martin Eigner. "An approach for the implementation of the digital twin in the automotive wiring harness field." DS 92: Proceedings of the DESIGN 2018 15th International Design Conference. 2018.
- [10] Kupriyanovsky, Vasily, et al. "BIM Technologies for Tunnels Used in Subways, Railways, Highways, and Hyperloop-IFC-Driven Real-Time Systems and Disruptive Innovation." International Journal of Open Information Technologies 8.9 (2020): 70-93.
- [11] Ganguli, R., and S. Adhikari. "The digital twin of discrete dynamic systems: Initial approaches and future challenges." Applied Mathematical Modelling 77 (2020): 1110-1128.
- [12] Law, Averill M., W. David Kelton, and W. David Kelton. Simulation modeling and analysis. Vol. 3. New York: McGraw-Hill, 2000.
- [13] Tang, Jiangjun, George Leu, and Hussein A. Abbass. Simulation and Computational Red Teaming for Problem Solving. John Wiley & Sons, 2019.
- [14] Nance, Richard E. "A history of discrete event simulation programming languages." History of programming languages---II. 1996. 369-427.
- [15] Moura, Ana Lúcia De, and Roberto Ierusalimsky. "Revisiting coroutines." ACM Transactions on Programming Languages and Systems (TOPLAS) 31.2 (2009): 1-31.

# Digital twins and discrete-event simulation systems

Dmitry Namiot, Oleg Pokusaev, Vasily Kupriyanovsky, Mikhail Zhabitskii

**Abstract—** This article is about the development tools for digital twins. Digital twins are getting more and more attention. At the same time, one cannot fail to note the fact that most of the publications consider the use of digital twins. How these twins were built (developed) is out of the question. Meanwhile, it is obvious that it is the questions of the development (creation) of twins that are primary. Any digital twin includes a model of a real object, presented in one form or another. This is what is called a digital replica. The paper deals with the relationship between systems for the development of discrete simulation models and digital twins. Discrete event modeling is one of the oldest and most widely used approaches to building modeling systems. The paper discusses add-ons for discrete-event modeling systems, which will allow the use of such tools for the development of digital twins. Such systems will also be important for universities in terms of training in the development of digital twins.

**Keywords—** digital twin, simulation.

## REFERENCES

- [1] Varga, András. "Discrete event simulation system." Proc. of the European Simulation Multiconference (ESM'2001). 2001.
- [2] Fishman, George S. Discrete-event simulation: modeling, programming, and analysis. Springer Science & Business Media, 2013.
- [3] El Saddik, Abdulmotaleb. "Digital twins: The convergence of multimedia technologies." IEEE multimedia 25.2 (2018): 87-92.
- [4] Khajavi, Siavash H., et al. "Digital twin: vision, benefits, boundaries, and creation for buildings." IEEE Access 7 (2019): 147406-147419.
- [5] Kurganova, Nadezhda, et al. "Digital twins' introduction as one of the major directions of industrial digitalization." International Journal of Open Information Technologies 7.5 (2019): 105-115.
- [6] Kupriyanovsky, Vasily, et al. "Digital twins based on the development of BIM technologies, related ontologies, 5G, IoT, and mixed reality for use in infrastructure projects and IFRABIM." International Journal of Open Information Technologies 8.3 (2020): 55-74.
- [7] Qi, Qinglin, et al. "Enabling technologies and tools for digital twin." Journal of Manufacturing Systems (2019).
- [8] Tao, Fei, et al. "Digital twin driven prognostics and health management for complex equipment." Cirp Annals 67.1 (2018): 169-172.
- [9] Tharma, Rajeeth, Roland Winter, and Martin Eigner. "An approach for the implementation of the digital twin in the automotive wiring harness field." DS 92: Proceedings of the DESIGN 2018 15th International Design Conference. 2018.
- [10] Kupriyanovsky, Vasily, et al. "BIM Technologies for Tunnels Used in Subways, Railways, Highways, and Hyperloop-IFC-Driven Real-Time Systems and Disruptive Innovation." International Journal of Open Information Technologies 8.9 (2020): 70-93.
- [11] Ganguli, R., and S. Adhikari. "The digital twin of discrete dynamic systems: Initial approaches and future challenges." Applied Mathematical Modelling 77 (2020): 1110-1128.
- [12] Law, Averill M., W. David Kelton, and W. David Kelton. Simulation modeling and analysis. Vol. 3. New York: McGraw-Hill, 2000.
- [13] Tang, Jiangjun, George Leu, and Hussein A. Abbass. Simulation and Computational Red Teaming for Problem Solving. John Wiley & Sons, 2019.
- [14] Nance, Richard E. "A history of discrete event simulation programming languages." History of programming languages---II. 1996. 369-427.
- [15] Moura, Ana Lúcia De, and Roberto Ierusalimsky. "Revisiting coroutines." ACM Transactions on Programming Languages and Systems (TOPLAS) 31.2 (2009): 1-31.
- [16] Simula <https://en.wikipedia.org/wiki/Simula> Retrieved: Jan, 2021.