

**Кудрин Ю.Ю.** Технический руководитель отдела разработки программного продукта. ООО Гедокорп РФ, г. Самара

## **ХРАНЕНИЕ И ПЕРЕДАЧА ЦИФРОВОГО ДВОЙНИКА МЕЖДУ СИСТЕМАМИ С ОТЛИЧАЮЩИМСЯ ФУНКЦИОНАЛОМ**

В работе проведено исследование особенностей использования цифрового двойника в условиях отсутствия «бесшовной интеграции», определены требования и рекомендации к структуре цифрового двойника, позволяющие обеспечить передачу данных между системами с отличающимся функционалом без роста его деградации. Исследование проведено методом теоретического анализа через моделирование ситуаций в рамках заданных ограничений и особенностей рассматриваемой системы.

The paper examines the features of using a digital twin in the absence of "seamless integration", defines the requirements and recommendations for the structure of a digital twin that allow ensuring data transfer between systems with different functionality without increasing its degradation. The study is carried out by the method of theoretical analysis through modeling situations within the given limitations and features of the system under consideration.

**Ключевые слова:** Цифровой двойник, структура данных, обмен данными, Content Management System (Система управления контентом), Enterprise Resource Planning (Планирование ресурсов предприятия), Manufacturing Execution System (Система управления производством), SPDM-система (Simulation Process & Data Management)

**Keywords:** Digital twin, data structure, data exchange, Content Management System, Enterprise Resource Planning, Manufacturing Execution System, SPDM system (Simulation Process & Data Management)

## ВВЕДЕНИЕ

«ЦД – это набор подходов и решений, призванных решить проблему, которая заключается в том, что рост сложности современных систем, а также многокомпонентных и многофункциональных продуктов опережает рост возможностей инструментов для их проектирования, изготовления и безопасного обслуживания. И решение этой проблемы лежит в объединении ряда цифровых технологий для предложения более эффективных средств моделирования, проектирования, создания и обслуживания подобных сложных систем. Эти инструменты должны комплексно описывать объект, бесшовно интегрироваться между собой, обеспечивая цифровую непрерывность среды создания продукта, работать в тесной информационной связке с предметом моделирования и отслеживать все фазы его развития (фазы жизненного цикла).»  
[2, с.29]

Цифровой двойник (далее ЦД) в современном мире решает задачу предоставления однозначной, частично абстрагированной, но в необходимой степени подробной информации об объекте или процессе.

Работа с ЦД проще и точнее, чем с непосредственным физическим источником данных, так как в случае ЦД информация уже локализована в одном месте и представлена в конкретных цифровых значениях. Для того, чтобы получить значение какого-либо из параметров объекта, нет необходимости подходить к нему с линейкой или иным измерительным инструментом.

Работа с ЦД проще и быстрее, чем с архивами документации. Нет необходимости искать паспорт изделия или иные документы для получения нужных данных по объекту. ЦД содержит все необходимые данные в себе.

Таким образом, логичным результатом эволюционного развития является переход от работы с отдельными документами к работе с ЦД и переход от хранения данных в виде отчетов и справок к хранению данных в виде однозначной модели.

«Ключевым событием в процессе перехода на MBSE является переход от документоориентированного подхода к моделиориентированному. В чем суть данного перехода? Большинство организаций давно полагаются на обмен документами в электронной форме. Это могут быть текстовые документы, таблицы Excel или чертежи в формате PDF. Но тот факт, что данные хранятся в электронных файлах, не означает, что они доступны в разных подсистемах проектирования изделия, поскольку для извлечения из этих файлов необходимой информации все еще требуется человек. Документ на естественном языке не предполагает извлечения однозначно определяемой информации» [2, с.110]

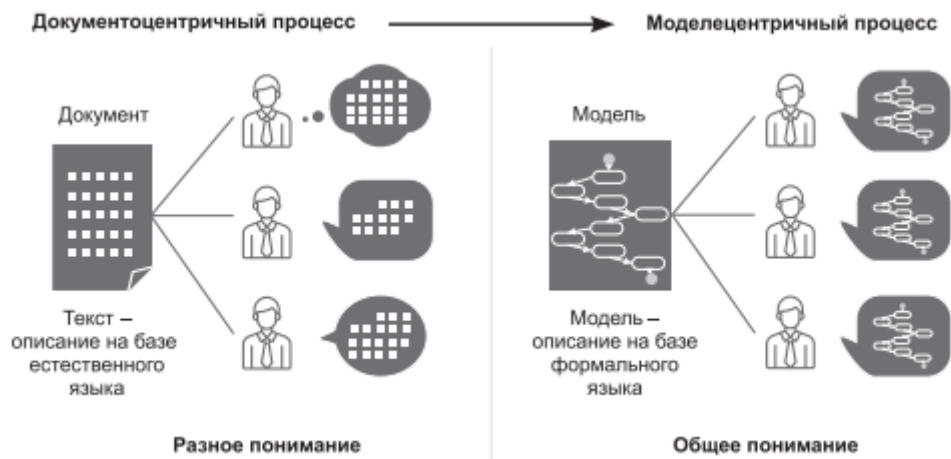


Рис.1 Сравнение документоцентричного и моделицентричного процессов. Источник Zhang, X

Сама логика и определение ЦД подразумевает глобализацию и централизацию данных. Идеальная система работы с ЦД подразумевает центральное хранилище данных, где этот двойник находится, и где происходит его непрерывное обновление с учетом информации, получаемой с измерительных приборов, регистрирующих состояние физического двойника. Также подразумевается доступность этого центрального хранилища для всех участвующих в работе сторон. Экстраполируя, мы можем с большой долей уверенности предположить, что в будущем будет построена подобная система под регулировкой государственных органов контроля, которая позволит получить одновременную доступность и единственность «источника истины»,

полную интеграцию данных различных отраслей и организаций друг с другом (в частности, единую систему справочников, каталогов и нормативов), и при этом обеспечит пропускной режим доступа, систему вознаграждений за предоставление данных и гарантированность наказаний за злоупотребления. Также, с большой долей уверенности, можно говорить, что до появления подобной системы и становления ее доступной для всех участников рынка, включая мелкий и средний бизнес, пройдет еще немало времени. Текущая же ситуация выглядит иначе.

*«Приложения моделирования инженерных подсистем должны быть интегрированы. То же самое справедливо и для платформы управления цепочкой создания стоимости, которая должна опираться на единую цифровую среду, на интегрированные данные по всей сети поставщиков. На практике, однако, ситуация часто выглядит иначе: существует большое число игроков – разработчиков дизайна, производителей оборудования, подрядчиков и т. д. Проблема моделирования и управления данными во всей цепочке создания стоимости оказывается сложной в силу отсутствия интеграции данных. Передача данных осуществляется в сети поставщиков, каждый из которых пользуется своей учетной системой» [2, с.130]*

На практике мы имеем множество участников процесса, у каждого из которых есть своя база данных, своя система учета, документооборота, свои системы взаимодействия с данными (всевозможные CMS, ERP, MBSE, даже банальный текстовый редактор или Excel). Эта ситуация является сдерживающим фактором для использования ЦД и для получения всех сопутствующих ему преимуществ. Фактически, на данный момент только крупнейшие корпорации могут эффективно использовать эту концепцию, так как одновременно и закрывают основную долю своих потребностей собственными мощностями, и имеют возможность обеспечить цельность информационного пространства внутри себя.

Данное исследование направлено на поиск решений, способных помочь развитию применения ЦД в менее крупных и более дифференцированных структурах, путем формулирования основных постулатов и ограничений, при использовании которых может быть выработан стандарт хранения и обработки данных, позволяющий минимизировать количество проблем при организации моделицентричного взаимодействия организаций с неидентичными системами обработки данных.

## **МЕТОДЫ**

Данное исследование является теоретическим.

Для получения выводов применяется метод логического анализа теоретической абстрактной системы с заданными ограничениями.

Для формулирования требований и рекомендаций производится анализ отдельных частных случаев в рамках рассматриваемой системы.

## **ИССЛЕДОВАНИЕ**

### **ГРАНИЧНЫЕ УСЛОВИЯ**

Зададим граничные условия теоретической абстрактной системы:

1. В системе имеется несколько акторов (организаций, отвечающих за свою отдельную область деятельности)
2. Имеется первичный актор, являющийся точкой начала процесса
3. Имеется конечный актор, являющийся точкой завершения процесса
4. В системе отсутствует единое информационное пространство (акторы не имеют общей централизованной системы данных, справочников и каталогов)
5. Между акторами имеются лимитированные каналы коммуникации (акторы могут осуществлять обмен данными в импульсном кратковременном режиме, например, в формате передачи файлов или текстовых сообщений, но возможность подключаться к системам других акторов через API отсутствует вследствие, например, нормативов информационной безопасности)

6. Актеры изменяют ЦД, работая каждый со своим блоком данных этого ЦД, которые частично пересекаются, но не обязательно повторяют границы блоков данных, которые используют иные актеры
7. На входе и выходе систем взаимодействия с данными (далее СВД; см. раздел «Термины») каждого актора данные должны представлять из себя ЦД
8. Каждый актер системы может получать или не получать данные непосредственно от физического двойника
9. База данных и справочники каждого актора системы конечны и не всеобъемлющи

Сформируем случайную систему, для отображения движения информации между актерами. Постараемся предусмотреть большинство возможных частных случаев (Рис.2).

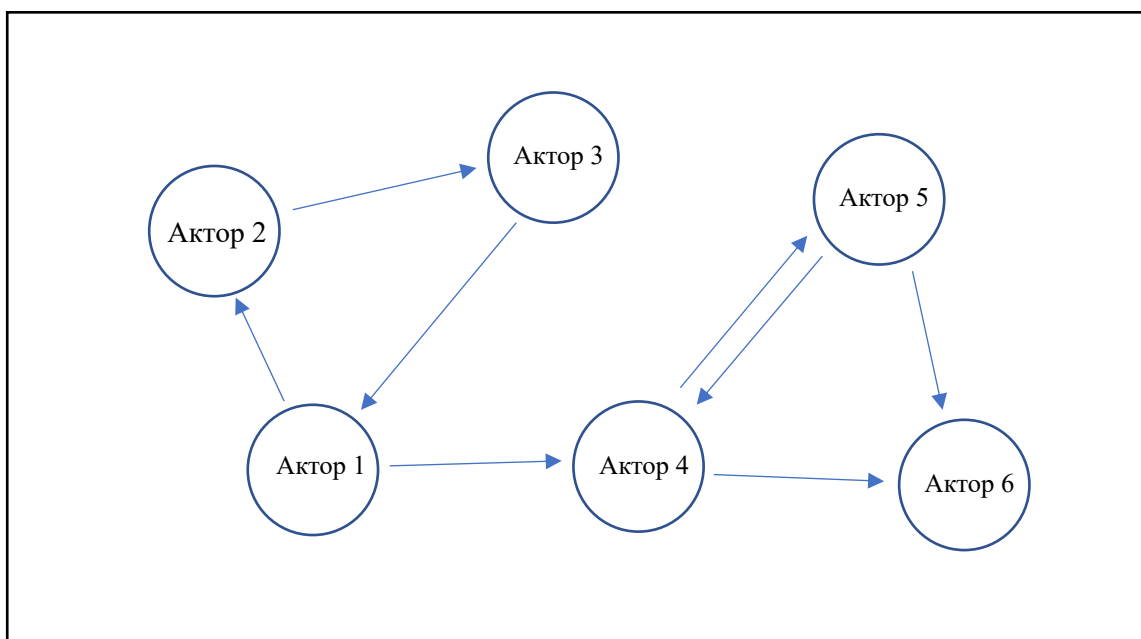


Рис.2 Актеры системы и движение информации между ними

На данной схеме первичный актер – это Актор 1, конечный актер – это Актор 6. Остальные актеры вносят свой вклад в процесс, изменяя и дополняя данные. Первичный актер формирует исходный ЦД. Конечный актер на выходе имеет свою версию ЦД, имеющую в себе данные от каждого из участников системы.

## ПЕРВИЧНЫЙ АНАЛИЗ. ФОРМУЛИРОВАНИЕ ЗАКОНОВ ХРАНЕНИЯ И ПЕРЕДАЧИ ЦИФРОВОГО ДВОЙНИКА МЕЖДУ СИСТЕМАМИ С ОТЛИЧАЮЩИМСЯ ФУНКЦИОНАЛОМ

Для удобства, введем термин «информационно-неоднородная система» и дадим ему определение:

*Информационно-неоднородная система (ИНС) в контексте работы с ЦД – это система, отдельные акторы которой не имеют единого информационного пространства и работают каждый со своими базами данных и системами обработки информации.*

Это определение соответствует всем принятым нами ограничениям и может быть использовано в дальнейших пояснениях.

Так как между акторами 1 и 6 нет прямой связи, а также вследствие ограничений 4 и 6, можно утверждать, что ЦД первичного актора и ЦД конечного актора не идентичны. Но, учитывая, что ЦД любого актора является результатом обработки ЦД актора 1, можно также утверждать, что все эти ЦД являются виртуальными двойниками одного и того же физического объекта. Изменяя данные ЦД, акторы меняют его адекватность.

*Адекватность модели – соответствие модели моделируемому изделию (процессу, явлению) по обоснованному перечню характеристик. [1, с.425]*

Учитывая выше сказанное, можно сформулировать первый закон хранения и передачи цифрового двойника между системами с отличающимся функционалом:

*Цифровые двойники различных акторов информационно-неоднородной системы имеют один, общий для всех, физический двойник, но при этом, как правило, не являются идентичными и обладают разным уровнем адекватности этому физическому двойнику.*

Теперь рассмотрим процесс перехода ЦД от одного актора к другому. Поскольку введено ограничение 4, очевидно, что нельзя обеспечить

непрерывность обновления данных ЦД от первичного актора. Но пункт 8 ограничений предполагает возможность наличия доступа к физическому объекту у любого актора системы. Давайте рассмотрим, что происходит при попытке поддерживать ЦД в режиме реального времени на уровне каждого отдельного актора.

Допущение, которое нам для этого потребуется, будет звучать следующим образом: любой актор может поддерживать состояние ЦД по всем его параметрам в режиме реального времени, получая данные непосредственно от физического объекта.

Это противоречит ограничению 6, заставляя каждого актора работать со всей информацией сразу, что, в свою очередь, требует от каждого актора системы оперировать всем спектром информации, необходимым для заполнения ЦД. Данное требование снижает перечень организаций, которые смогут обеспечить реализацию подобного подхода, что является существенным минусом. Также это создает риски некорректного внесения данных, так как по условиям, каждый актор системы специализируется на чем-то своем и, как следствие, не обладает достаточными компетенциями, чтобы корректно вносить и изменять данные за пределами своей области. Таким образом, можно сделать предварительный вывод о неправомерности высказанного допущения.

Тем не менее, попробуем разобраться эту ситуацию подробнее, для подтверждения или опровержения этого вывода. Рассмотрим ситуацию, когда путь движения ЦД закольцовывается, и данные возвращаются к источнику. На схеме есть два подобных кольца, которые формируются стрелками 1,2 и 3, а также 5 и 6. Для того чтобы описать возникающую в данном случае проблему, необходимо сформулировать причины, по которым осуществляется передача ЦД к другим акторам системы. Очевидно, что если бы актор мог своими силами дополнить данные ЦД, то не возникало бы необходимости передачи данных другому актору системы.

Следуя от противного, можем сформулировать второй закон хранения и передачи цифрового двойника между системами с отличающимся функционалом:

*В рамках ИНС отдельные акторы имеют каждый свою специализацию. Передача ЦД между акторами осуществляется для дополнения ЦД данными, относящимися к специализации актора, принимающего его для редактирования.*

Продолжим рассмотрение ситуации, когда ЦД возвращается к актору-источнику. Согласно первому закону, ЦД, который хранит и поддерживает актор-источник, не будет эквивалентен ЦД, возвращающемуся от внешних акторов. Учитывая невозможность идеального согласования приборов измерения разных акторов, можно гарантировать несовпадение данных в той или иной степени по всему объему ЦД. Возникает классический конфликт слияния разных версий. Рассмотрим варианты разрешения этого конфликта:

1. Отбрасываются все изменения внешнего актора, кроме дополнений.
2. Принимается входящая версия ЦД со всеми изменениями.
3. Принимаются все изменения, кроме той части, которая относится к специализации текущего актора.
4. Принимается та часть изменений, которая относится к специализации внешнего актора.

Метод 1: отбрасываются все изменения внешнего актора, кроме дополнений.

В этом случае принимающий актор считает, что его ЦД адекватнее, и только расширяет его данными, полученными извне. То есть принимаются только те данные, которые записываются в пустующие позиции.

В результате теряется часть информации, которая относится к компетенции внешнего актора, и в которой он является большим специалистом, чем текущий

актор, если эта информация перекрывает ранее заполненные поля. Это лишает передачу ЦД внешнему актору для заполнения этих данных смысла.

Вывод: первый метод нарушает логику системы и не должен применяться в ее рамках.

Метод 2: принимается входящая версия ЦД со всеми изменениями.

В этом случае принимающий актер считает, что его версия ЦД устарела, и принимает все изменения полностью. Таким образом он затирает, в том числе, те блоки, которые заполнялись в рамках его компетенций или в рамках компетенций предыдущих акторов, заменяя их данными, предоставленными менее компетентными в данных областях специалистами.

Вывод: второй метод уменьшает релевантность данных, хранимых в ЦД, тем самым уменьшая его адекватность.

Метод 3: принимаются все изменения, кроме той части, которая относится к специализации текущего актора.

В этом случае, как и во втором, актер считает, что его ЦД устарела, но учитывает превалирование своих компетенций в заданной области. Таким образом, принимается все, кроме данных, которые находятся в его «зоне ответственности». В этом случае происходит снижение релевантности данных в блоках, которые находятся вне компетенции обоих акторов, которые формировались предыдущими актерами системы. К тому же стоит учитывать частный случай, когда происходит частичное пересечение компетенций, в результате которого будут отринуты те изменения, которые должны были быть применены.

Вывод: третий метод уменьшает релевантность данных, хранимых в ЦД, тем самым уменьшая его адекватность.

Метод 4: принимается та часть изменений, которая относится к специализации внешнего актора.

В этом случае текущий актер принимает только те данные, в которых компетентен передающий актер. Таким образом в итоге получается ЦД, в котором блоки текущего и передающего актора заполнены корректными

данными, сформированными специалистами в этой области. Но если мы рассматриваем кольцо 1-2-3, то становится очевидно, что будут утеряны блоки данных, сформированные (в данном примере) актором 2, так как на этапе актор3-актор1 квалификация актора 2 не учитывается.

Добавим в условие учет специализации всех акторов «кольца» и будем принимать те блоки, компетенция в которых есть хотя бы у одного из акторов. Результат становится более релевантным. Но тем не менее, в каждой точке кольца имеет место деградация данных, так как согласно допущению, каждый актор производит непрерывное обновление данных в ЦД, в том числе за пределами своей области компетенций.

Выводы:

1. Четвертый метод дает приемлемые результаты слияния данных, если учитывать компетенции всех участников кольца.
2. Имеет место деградация информации на каждом этапе движения данных между акторами.

Второй вывод имеет место даже если мы рассматриваем движение информации не по кольцу, а линейно. Таким образом, предварительный вывод о том, что допущение, сделанное вначале, ведет к уменьшению адекватности ЦД, подтверждается.

Проанализировав полученные выводы, можно сформулировать третий закон хранения и передачи цифрового двойника между системами с отличающимся функционалом:

*Каждый актор в ИНС может изменять только те данные, которые относятся к его компетенции, сохраняя прочие данные в неизменном виде.*

При соблюдении третьего закона мы получим в роли передаваемого по ИНС цифрового двойника статичный снимок, который выборочно модифицируется на каждом этапе пути. Очевидно, что это накладывает ограничения на применимость подобного подхода к процессам, в которых

цифровой двойник должен получать частые регулярные обновления всех данных. Тем не менее, имеется довольно большой набор процессов, в которых ЦД обновляется достаточно редко, чтобы сохранялся смысл передачи данных между акторами системы в виде статичного, частично модифицируемого снимка ЦД.

Сформулируем четвертый закон хранения и передачи цифрового двойника между системами с отличающимся функционалом:

*Хранение и передача цифрового двойника между системами с отличающимся функционалом должны применяться в основном для процессов с периодом обновления данных большим, чем время, необходимое для полного завершения цикла движения двойника по ИНС.*

## **МОДЕЛИРОВАНИЕ ПРОЦЕССА ДВИЖЕНИЯ ЦД ПО ИНС**

### **ОПРЕДЕЛЕНИЕ ОСНОВНЫХ МОМЕНТОВ ПРОЦЕССА**

Рассмотрим момент получения актором ЦД от предыдущего актора. Какие действия ему необходимо предпринять до передачи данных следующему актору?

1. Выделить идентификатор ЦД для идентификации его в системе и создания связей между относящимися к нему записями
2. Выделить блоки данных, с которыми будет проводиться работа
3. Разместить информацию из этих блоков данных в своей системе
4. Произвести обработку и редактирование данных внутри своей системы
5. Произвести формирование обновленного ЦД с использованием отредактированных данных, без искажения «незнакомых» блоков данных.

Выделение идентификатора и блоков данных по каким-либо признакам в теле ЦД вызывает необходимость соблюдения следующего требования:

*Структура данных ЦД должна быть формализована в едином для всех акторов виде.*

Можем ли мы продолжать говорить про информационно-неоднородную систему, если вводим общий формализованный принцип организации данных ЦД? Да, можем, пока формализация процесса не заходит далее описания блоков структуры ЦД, так как продолжает сохраняться принцип отсутствия единого информационного пространства для отдельных акторов системы.

Подобная организация обмена данными подразумевает наличие парсера-преобразователя у актора для каждой отдельной ИНС, в которой он принимает участие. При отсутствии возможности получить единое информационное пространство для всех систем, в которых актор принимает участие, данное решение может быть принято в качестве «необходимого зла» для получения возможности использовать моделицентричный процесс.

Основной минус, который несет данное требование – любая организация работы ИНС по этому принципу потребует подготовительных работ по созданию и интеграции парсера-преобразователя, осуществляющего размещение данных ЦД в таблицах СВД актора и формирование обновленного ЦД для передачи его следующему актору.

Плюсы, которые получают акторы при использовании моделицентричного процесса – это все плюсы работы с ЦД, в частности, возможность полной автоматизации всех процессов и, как следствие, минимизации человеческого участия в достижении результата. Они станут доступны сразу после настройки и согласования справочников с ИНС через парсер-преобразователь.

#### СПОСОБ ХРАНЕНИЯ И ПЕРЕДАЧИ ЦД

Учитывая вышеизложенное требование, а также все ограничения и сформулированные законы, можно утверждать, что ЦД должен передаваться в виде пакета данных (например, файла или архива с файлами) со структурированной информацией. При этом, данные внутри этого пакета должны быть сгруппированы по функциональному признаку, чтобы позволить актору выбрать те блоки данных, которые относятся к его области компетенций. Исходя из последнего, можно сформулировать второе требование к структуре данных ЦД:

*Данные цифрового двойника должны быть разделены на отдельные блоки по функциональному признаку. Эти блоки должны иметь однозначный, согласованный между всеми акторами системы идентификатор.*

Учитывая ограничение 6, можно утверждать, что ЦД содержит ряд блоков данных, не входящих в зону компетенций и интересов конкретного актора. Таким образом логичным будет следующее требование:

*Блоки данных должны быть ограничены фиксированными, однозначными в рамках ИНС идентификаторами с начала и конца блока.*

Хорошим вариантом для записи монофайлового ЦД, с учетом данного требования, будет xml структура или ее аналоги. Открывающий и закрывающий теги позволяют однозначно выделить известные блоки данных и изъять их для дальнейшей обработки, не повредив ту часть ЦД, которая выходит за границы компетенций текущего актора.

Если используется многофайловая структура, то в роли ограничивающих идентификаторов могут выступать сами границы файла. В этом случае каждый блок данных должен храниться в отдельном файле с соответствующим названием.

У каждого из этих подходов есть свои плюсы и минусы. В частности, многофайловая структура позволяет передавать лишь часть данных или компоновать ЦД вручную силами оператора, что может привести к нарушению ЦД. Монофайловая структура тоже может быть изменена вручную, но сделать это случайно или задать процесс ручной коррекции как стандартный будет сложнее. С другой стороны, при больших размерах ЦД хранение в монофайле может вызвать проблемы с производительностью.

В любом случае, окончательный выбор способа хранения и передачи ЦД должен определяться требованиями системы, в которой он будет применяться.

## ВЫДЕЛЕНИЕ ИДЕНТИФИКАТОРА ЦД

Очевидно, что для того, чтобы идентифицировать ЦД в рамках ИНС, данный идентификатор должен быть уникальным вне рамок СВД отдельного актора. Запишем это в виде требования к структуре данных ЦД.

*Цифровой двойник должен иметь глобально уникальный идентификатор (GUID)*

В качестве GUID можно использовать хэш-ключ, сгенерированный с использованием алгоритма хеширования SHA-3 на основании Unix-времени формирования единицы данных, внутреннего идентификатора в СВД и индивидуального ключа самой СВД, осуществляющей формирование этой единицы данных. Подобный подход обеспечит высокую степень уникальности идентификатора единицы данных, в том числе, при массовом заполнении справочников, позволяя однозначно идентифицировать ЦД в любой системе.

## ВЫДЕЛЕНИЕ НЕОБХОДИМЫХ АКТОРУ БЛОКОВ ДАННЫХ

Блоки данных должны быть прописаны при формализации структуры данных ЦД. Для их идентификации стоит использовать строковое обозначение, содержащее соответствующий блоку смысл. Например, «Finance», «Comments», «Points» и т.п.

Проводя размещение ЦД в своей СВД, актор должен отделить необходимые ему блоки данных для дальнейшего парсинга и обработки. Остальное тело ЦД должно быть сохранено в каком-либо виде для присоединения к пересозданному ЦД, во избежание снижения адекватности.

Если работа идет с многофайловой формой хранения ЦД, достаточно сохранить исходный пакет данных и финальным действием обновить те файлы, которые содержат нужные блоки данных. Если речь идет о монофайловой структуре, можно пойти таким же путем, подменяя в исходном файле нужные разделы, или сохранить все незнакомые блоки ЦД в виде строки во внутренней базе данных актора до тех пор, пока не наступит время сборки обновленного

ЦД. При этом сохраненные незнакомые блоки достаточно будет просто добавить в конец нового файла.

## ОБРАБОТКА ДАННЫХ ВНУТРИ СИСТЕМЫ И ФОРМИРОВАНИЕ ОБНОВЛЕННОГО ЦД

Основная задача, которая стоит перед актором – это размещение данных ЦД в таблицах базы данных своей внутренней системы. Эта задача должна решаться вышеупомянутым парсером-преобразователем. После этого, при корректной организации процесса, эти данные могут обрабатываться стандартным для системы образом. По завершении обработки эти данные должны быть преобразованы обратным преобразователем в ЦД такого же вида, что поступил на вход. После этого ЦД готов для передачи его к следующему актору.

Сформулируем проблемы, которые могут возникнуть в этих процессах.

1. Удаление связанных данных из неизвестных модулей
2. Добавление новых данных, не имеющих отражения в неизвестных модулях
3. Одновременная работа нескольких акторов над одним ЦД

Рассмотрим первую проблему. Попробуем смоделировать ситуацию. Пусть ЦД содержит перечень элементов. Пусть отдельный блок данных хранит финансовую информацию, связанную с этими деталями. В этом случае возможны две ситуации:

1. Производится удаление основной единицы данных
2. Производится удаление дополнительной единицы данных, связанной с основной единицей данных

В нашем смоделированном примере элемент перечня является основной единицей данных. Финансовые показатели, связанные с ним, будут дополняющей единицей данных, связанной с элементом. Очевидно, что, если мы удаляем из перечня элемент, желательно вычистить все дополняющие данные, которые были с ним связаны, как вторичные. Если же работа ведется с финансовым блоком информации и по какой-то причине требуется удалить

единицу данных оттуда, то сам элемент трогать не нужно, так как финансовый блок дополняет перечень элементов, но не является для него «опорным».

Можно сделать вывод, что для второй ситуации не требуется предпринимать никаких дополнительных действий. Для первой же ситуации, необходимо установить связь между основной единицей данных и дополняющей единицей данных (элементом и финансовыми параметрами, привязанными к нему, в нашем примере). Значит, должно существовать поле-идентификатор, обеспечивающее связь между данными разных блоков, относящихся к одной единице данных, и эта связь может быть однонаправленной.

Для получения однозначности ситуации при определении основной и дополняющей единиц данных, необходимо конкретизировать это требование в следующей форме:

*Единицы данных, относящиеся друг к другу, но находящиеся в разных блоках данных, должны быть связаны между собой идентификатором основной единицы. Поле-идентификатор основной единицы данных должно иметь стандартное для всего ЦД обозначение и хранить идентификационные данные. Связующее поле дополняющей единицы данных должно иметь стандартное для всего ЦД обозначение и хранить идентификатор блока данных основной единицы данных и идентификационные данные самой основной единицы данных.*

При исполнении данного требования появляется возможность при удалении единицы данных провести поиск по прочим, в том числе неизвестным блокам данных, для обнаружения единиц данных с соответствующим идентификатором и удаления их. Для сохранения целостности данных необходимо рекурсивно проверять ЦД на единицы данных, дополнительные для единиц данных, удаляемых таким образом. Исполнение вышеизложенного требования приведет к созданию структуры, позволяющей данное действие в том числе в неизвестных блоках данных.

Рассмотрим вторую проблему. Мы можем утверждать, что либо добавляемая единица данных является основной, а не дополняющей единицей данных, либо актер имеет доступ к блоку данных, в котором располагается основная единица данных, которую дополняет добавляемая единица данных. Логично предположить, что добавление дополняющей информации без наличия основы для нее не несет смысла, а значит, наше утверждение истинно. Значит, вторая проблема сводится к невозможности создать дополняющую информацию в неизвестных модулях. В рамках рассматриваемой системы эта ситуация не решаема до перехода к актору, обладающему компетенциями изменять те блоки данных, в которые должна быть внесена эта дополняющая информация. Но можно утверждать, что до этого момента и необходимости в этой информации также не существует.

Третья проблема, связанная с одновременной работой нескольких акторов над одним ЦД, может возникнуть у актора 6 на рис.2. Данные к нему поступают от акторов 4 и 5. По первому закону, получаемые актором 6 два варианта ЦД будут не идентичны. Рассмотрим возможные ситуации:

1. Предыдущие акторы обладают непересекающимися компетенциями
2. Предыдущие акторы обладают одинаковыми компетенциями
3. Предыдущие акторы обладают частично пересекающимися компетенциями

Первая ситуация является идеальной. В ней достаточно извлечь из ЦД одного актора те блоки данных, в которых он компетентен, и внедрить их в ЦД второго актора, для получения одного «актуального» ЦД, подходящего для дальнейшей работы.

Вторая ситуация является полностью негативной. Если мы можем предположить, что акторы обладают одинаковым уровнем компетентности, тогда нужно взять более новый вариант ЦД. Если уровень компетентности разный, тогда нужно взять ЦД более компетентного актора. В обоих вариантах, один из акторов игнорируется, а значит система имеет нарушение внутренней логики.

Третья ситуация сложнее. Очевидно, что непересекающиеся блоки данных должны быть взяты каждый у соответствующего актора. Но все равно возникает вопрос общих блоков. В данной ситуации нет однозначно правильного решения вопроса, так как в общем блоке могут быть расположены единицы данных, являющиеся основными для дополняющих единиц данных в непересекающихся блоках, что приводит к невозможности отдать приоритет версии одного из акторов и требует аккуратного слияния данных. Слияние возможно сделать только в том случае, если принимающий актор (актор б в нашем примере) имеет компетенции в сливаемом блоке данных. Во всех остальных случаях безошибочный вариант действий отсутствует.

Таким образом можно сформулировать рекомендацию:

*Рекомендуется избегать ситуаций, в которых над одним ЦД выполняют работы акторы с пересекающимися компетенциями, во избежание конфликтов данных. Если такие ситуации имеют место, то каждая единица данных должна иметь поле с датой последних изменений для разрешения конфликтов изменения данных, и глобально уникальный идентификатор для разрешения конфликта при одновременном добавлении на один идентификатор новых единиц данных разными акторами.*

## **ТИПОВЫЕ ЕДИНИЦЫ ДАННЫХ. ИНТЕГРАЦИЯ ДАННЫХ В СПРАВОЧНИКИ РАЗЛИЧНЫХ АКТОРОВ ИНС**

### **АНАЛИЗ ВХОДЯЩИХ ЕДИНИЦ ДАННЫХ. ОПРЕДЕЛЕНИЕ ЗАДАЧ, ТРЕБУЮЩИХ РЕШЕНИЯ ПРИ РАБОТЕ С ТИПОВЫМИ ЕДИНИЦАМИ ДАННЫХ**

Рассмотрим частный случай: ЦД содержит блок данных, сформированный из данных типовых элементов, изделий или параметров, которые сведены в справочник, и этот блок используется двумя или большим числом акторов.

Вследствие того, что система у нас неоднородна, и вследствие первого закона можно утверждать, что для всех акторов после первого данный блок будет восприниматься «чужеродным», то есть не связанным пространством идентификаторов. Например, если мы говорим о ЦД некоего изделия, состоящего из стандартных комплектующих, и блок данных, который мы

рассматриваем - это его состав, то мы получаем ситуацию, в которой условная «Гайка М8» в справочнике исходного актора изначально не будет связана с аналогичной записью «Гайка М8» в справочнике других акторов через базовый идентификатор, формируемый итерацией. При этом у части акторов этой гайки может вообще в справочниках не оказаться.

Таким образом, нужно решить несколько задач:

1. Однозначность идентификации типовой единицы данных в этом и прочих ЦД различными акторами системы
2. Интеграция новых типовых единиц данных в справочники акторов
3. Связь типовых единиц данных внешних акторов с их аналогами во внутренних справочниках конкретного актора

Первая задача, очевидно, решается, если дополнить требование к структуре данных ЦД в ИНС, сформулированное ранее, приведя его к следующему виду:

*Цифровой двойник и все единицы данных, являющиеся элементами справочников и обладающие параметрами, определяемыми в этих справочниках, должны иметь глобально уникальный идентификатор (GUID)*

Примеры единиц данных, для которых требуется GUID:

- Комплектующие (такие как балки, изделия, крепеж)
- Материалы по ГОСТ, если для них указаны дополнительные параметры, такие как твердость, масса и т.п.
- Строительные объекты (здания, котлованы, мосты и т.п.)

Для поиска решений остальных задач рассмотрим процессы, возникающие при анализе новой ЦД, полученной актором.

#### ИНТЕГРАЦИЯ ТИПОВЫХ ЕДИНИЦ ДАННЫХ В СПРАВОЧНИКИ СВД И СЛИЯНИЕ ЭКВИВАЛЕНТНЫХ ДАННЫХ

Следующее действие, которое должно производиться с единицей данных после ее выделения из тела ЦД – добавление ее в соответствующий справочник в виде новой записи. Для этого необходимо разнести данные из ЦД по правильным полям справочника СВД текущего актора.

Для решения этой задачи, необходимо ввести классификацию для параметров единицы данных:

- **Идентификационные параметры** – это параметры, обеспечивающие однозначную идентификацию единицы данных среди однотипных с ней единиц. *Примером такого параметра будет GUID или идентификатор, имеющий смысл во внутреннем пространстве данных ЦД.*
- **Описательные параметры** – это параметры, исполняющие функцию именования единицы данных с использованием общепринятых смыслов и понятий. В контексте хранения и передачи цифрового двойника между системами с отличающимся функционалом, описательные параметры выполняют задачу определения эквивалентности незнакомой единицы данных аналогам в справочниках актора. *Примером такого параметра будет являться «Наименование», «Артикул» или схожие идентификаторы единицы, если они являются номером в некой глобальной базе данных, как, к примеру, «ИНН» является номером в государственных реестрах.*
- **Качественные параметры** – это параметры, хранящие или олицетворяющие конкретные измеримые величины, имеющие однозначный математический или физический смысл. Как правило, выражаются цифрами. *Примером такого параметра может являться масса, стоимость или материал (если последний имеет соответствие в справочнике материалов, где указанному идентификатору соответствует, например, химический состав с точными процентами составляющих).*
- **Внутренние параметры** – это параметры, относящиеся к внутренней логике системы и не несущие смысла вне ее. Примером такого параметра может являться идентификатор во внутренней базе

данных СВД актора или «Артикул» единицы, если он используется исключительно внутри системы.

Некоторые параметры могут объединять признаки описательных и качественных параметров. Например, ГОСТ или ТУ изделия. Они являются качественными параметрами, так как за ними лежат конкретизированные, измеримые физические требования, но также несут и описательный характер, дополняя наименование единицы данных для определения аналога среди множества схожих.

Очевидно, что идентификационные параметры и описательные параметры имеют схожую функцию на разных этапах обработки данных. Идентификационные параметры позволяют автоматически идентифицировать знакомую единицу данных. Описательные параметры предназначены для идентификации незнакомой единицы данных и поиска ее эквивалента в справочниках актора. Из этого можно вывести следующее требование к структуре данных ЦД:

*Для типовой единицы данных, кроме уникального набора значений идентификационных параметров (GUID), должен иметь место уникальный набор значений описательных параметров.*

Таким образом, если в справочнике у актора имеется несколько записей с одинаковым наименованием «гайка», то в ЦД, кроме этого поля, должны быть использованы дополнительные описательные поля, гарантирующие уникальность данной типовой единицы данных относительно прочих типовых единиц. Пример корректной структуры данных:

- GUID – уникальный хэш-ключ, соответствующий единственной записи в справочнике актора
- Name – «Гайка»
- Type – «М8» (если в справочнике имеется несколько записей с Name = «Гайка»)

- GOST – «ГОСТ 5915-70» (если в справочнике имеется несколько записей с Name = «Гайка» и Type = «М8»)

Возвращаемся к озвученной задаче – внести данные из ЦД в соответствующие поля.

Справочники различных акторов могут быть организованы по самым разным принципам. Запись, соответствующая гайке М8, выполненной по ГОСТ 5915-70, может быть представлена в виде нескольких полей («Гайка» + «М8» + «ГОСТ 5915-70») или в виде одного поля («Гайка М8 ГОСТ 5915-70»), или в виде одного поля с иным образом организованной информацией («Гайка 5915-70 м8 железная», «гайка м8», «гайка» и т.п.).

Таким образом, возникает требование по конвертации данных описательных полей в формат, стандартный для СВД текущего актора. Можно утверждать, что высокая степень дискретизации описательных полей является более удобной для конвертирования, так как собрать алгоритмом из «Гайка», «М8», «ГОСТ 5915-70» одно составное название можно легко, а провести парсинг составного названия на отдельные поля гораздо сложнее. На основании этого можно сформулировать следующую рекомендацию:

*Структуру описательных параметров единицы данных рекомендуется выстраивать с позиции максимальной дифференциации по смыслу (форма, тип, материал, стандарт, размер и т.п.)*

Для оптимизации объема ЦД имеет смысл ввести отдельные блоки данных - справочники, задачей которых будет являться идентификация отдельных типовых единиц данных. Это позволит хранить детальную описательную информацию только в одном блоке данных ЦД, в прочих же - использовать только GUID типовых единиц данных.

Теперь рассмотрим третью задачу.

Имеется частный случай, когда добавленная новая единица данных имеет свой эквивалент в личных справочниках актора. Есть два варианта разрешения этой ситуации:

1. Оставить новую единицу данных в справочнике отдельной записью
2. Провести поиск аналогов и слияние нескольких записей в одну

Если экстраполировать первый вариант на бесконечность, можно утверждать, что в перспективе длительной эксплуатации системы он приведет к перегрузке базы данных двойниками. Из чего однозначно следует, что его нельзя рекомендовать к использованию.

Второй вариант лишен этого недостатка. Он не ведет к чрезмерному разрастанию баз данных и не затрудняет работу оператора поиском среди множества «гаек м8» нужной. Стоит отметить, что слияние единиц данных, взятых из справочников разных акторов, является процессом синхронизации этих справочников, что ведет к частичному выполнению требования по бесшовной интеграции, который заявляется для идеальной формы использования ЦД. Это косвенно подтверждает правильность второго варианта. Но подобный подход подразумевает слияние данных разных акторов, а значит, накладывает ряд ограничений. Проведем анализ ситуации и сформулируем эти ограничения.

Прежде всего, очевидно, что у разных акторов могут оказаться разные наборы полей для однотипных справочников. Например, у одного актора «гайка м8» кроме названия содержит «артикул», «гост» и «материал», а у другого вместо этих трех будут поля «срок поставки» и «стоимость».

Рассмотрим пример с гайками подробнее. Исходный актор внес в ЦД упоминание «гайки». В его справочниках для данного типа единицы данных использованы, в числе прочих, следующие поля: «артикул», «ГОСТ», «Резьба». При этом имеется N гаек с одинаковым названием, но отличиями в прочих параметрах. В спецификации ЦД структура единицы данных, во исполнение рекомендации по дифференциации, содержит поля «ГОСТ» и «Резьба». Артикул, в данном случае, можно рассматривать как описательный параметр, только если он является номером в каком-либо внешнем реестре. В противном случае, он является внутренним параметром и не несет смысла вне СВД

конкретного актора. При указанной структуре, принимающий актер может оценить разницу между разными гайками, упоминающимися в ЦД.

Рассмотрим случай, когда принимающий актер имеет идентичную структуру справочника. Очевидно, что при полном совпадении описательных параметров (с приведением к единому регистру), имеет смысл провести слияние новой единицы данных с уже имеющейся эквивалентной. Для этого необходимо завести в СВД структуру, отслеживающую ассоциации внешних GUID и внутренних GUID. Например, для этого можно использовать кросс-таблицу, где первый столбец будет хранить GUID справочников актора, а второй столбец – GUID ассоциированных ему эквивалентов извне.

Рассмотрим другой вариант. Актор, принимающий ЦД, в своих справочниках имеет следующие поля для описания той же гайки: «наименование», «GUID», «срок поставки» и «стоимость». И при этом у него в справочнике есть только одна запись для гайки, которая имеет наименование «гайка с 8ой резьбой». Очевидно, что при подобной организации именовании записей получить ассоциацию при помощи алгоритмических методов очень сложно. Остается либо использование нейросетей, либо слияние в ручном режиме, либо добавление новой записи в справочник.

Можно сделать вывод, что решение третьей задачи осуществимо только при системной и грамотной организации справочников актора. В противном случае придется проводить ручное слияние каждый раз, когда встретится новый GUID для имеющегося элемента, или перегружать справочники клонами.

### **ПРОЦЕСС ОБРАТНОЙ СБОРКИ ЦД**

Рассмотрим процесс обратной сборки ЦД после обработки в СВД актора. Как мы определили выше, при формировании обновленной структуры необходимо использовать GUID исходного актора. Это усложнит процесс формирования ЦД, но предотвратит потенциальные ошибки и возможную деградацию ЦД.

Рассмотрим варианты решения данной задачи.

Добавление всех возможных вариантов GUID, относящихся к данной единице данных, может быть простым решением в данном случае, но, если провести экстраполяцию на бесконечность, мы увидим неконтролируемое разрастание ЦД, что является неприемлемой ценой за упрощение формирования ЦД акторами.

Принятие единого GUID после слияния данных не является приемлемым решением. Конкретные единицы данных могут быть участниками процессов с разными ЦД, полученными от разных акторов. Соответственно, это вызовет периодическую смену GUID без возможности выработать общую базу.

Сохранение исходного тела ЦД и формирование индекса ассоциаций GUID справочников текущего актора и ЦД перед формированием обновленного ЦД ведет к высоким расчетным нагрузкам на систему, но позволяет решить поставленную задачу без деградации или иного ухудшения ЦД или СВД актора.

#### **ВАРИАНТЫ ПРИМЕНЕНИЯ ЦД В ИНС**

Пример 1:

- Предприятие хранит свои системы коммуникаций в виде цифровых двойников.
- Для проведения диагностики, ЦД передается камеральной организации, которая обеспечивает снятие замеров и фиксацию данных в ЦД.
- Камеральная организация передает ЦД монтажной организации, обеспечивающей подготовительные работы и фиксирующей в ЦД расстановку строительных лесов, их статус и элементы коммуникаций, подготовленные к замерам.
- Монтажная организация проводит периодическую передачу обновленных данных ЦД в камеральную организацию, до полного завершения цикла работ.
- Предприятие получает обновленный ЦД с результатами замеров и данными по точкам монтажа строительных лесов с указанием типа

конструкций и данных по количеству комплектующих, что позволит сократить времязатраты на следующий цикл диагностики.

Пример 2:

- Энергоснабжающая организация хранит данные по своим линиям и оборудованию в виде цифровых двойников.
- Эта организация имеет ряд различных подрядчиков, отвечающих за обслуживание своих участков, не объединенных в единое информационное пространство.
- В процессе обслуживания, подрядчики производят периодическую выгрузку данных по своим участкам в центральную компанию.
- В случае ремонта подрядчик передает ЦД ремонтной организации.
- Ремонтная организация дополняет ЦД данными по ремонту, передавая данные подрядчику.
- Завод получает обновленный ЦД, содержащий данные по текущему ремонту и состоянию линий, позволяющий производить статистический анализ и прогнозирование в автоматическом режиме.

## **ВЫВОДЫ**

Теоретическое исследование и моделирование различных ситуаций показали принципиальную реализуемость передачи цифрового двойника между системами с отличающимся функционалом и помогли сформулировать базовые законы, требования и рекомендации для данного процесса.

В результате исследования сформулированы четыре закона, которые должны исполняться при хранении и передаче цифрового двойника между системами с отличающимся функционалом:

1. Цифровые двойники различных акторов информационно-неоднородной системы имеют один, общий для всех, физический двойник, но при этом, как правило, не являются идентичными и

обладают разным уровнем адекватности этому физическому двойнику.

2. В рамках ИНС отдельные акторы имеют каждый свою специализацию. Передача ЦД между акторами осуществляется для дополнения ЦД данными, относящимися к специализации актора, принимающего его для редактирования.
3. Каждый актор в ИНС может изменять только те данные, которые относятся к его компетенции, сохраняя прочие данные в неизменном виде.
4. Хранение и передача цифрового двойника между системами с отличающимся функционалом должны применяться в основном для процессов с периодом обновления данных большим, чем время, необходимое для полного завершения цикла движения двойника по ИНС.

В результате исследования сформулированы следующие обязательные требования к структуре цифрового двойника:

1. Структура данных ЦД должна быть формализована в едином для всех акторов виде.
2. Данные цифрового двойника должны быть разделены на отдельные блоки по функциональному признаку. Эти блоки должны иметь однозначный, согласованный между всеми акторами системы идентификатор.
3. Блоки данных должны быть ограничены фиксированными, однозначными в рамках ИНС идентификаторами с начала и конца блока.
4. Цифровой двойник и все единицы данных, являющиеся элементами справочников и обладающие параметрами, определяемыми в этих справочниках, должны иметь глобально уникальный идентификатор (GUID)

5. Единицы данных, относящиеся друг к другу, но находящиеся в разных блоках данных, должны быть связаны между собой идентификатором основной единицы. Поле-идентификатор основной единицы данных должно иметь стандартное для всего ЦД обозначение и хранить идентификационные данные. Связующее поле дополняющей единицы данных должно иметь стандартное для всего ЦД обозначение и хранить идентификатор блока данных основной единицы данных и идентификационные данные самой основной единицы данных.
6. Для типовой единицы данных, кроме уникального набора значений идентификационных параметров (GUID), должен иметь место уникальный набор значений описательных параметров.

В результате исследования сформулированы следующие принципы взаимодействия с цифровым двойником, содержащим неизвестные блоки данных, обязательные для любых систем, взаимодействующих с этими цифровыми двойниками в рамках описываемой парадигмы:

1. СВД должна иметь возможность назначить для записи в своей таблице данных дополнительные идентификаторы для связи собственных справочников с данными импортируемых ЦД.
2. Неизвестные блоки данных цифрового двойника должны сохраняться при экспорте целиком (например, в виде строки), и должны прикрепляться к телу цифрового двойника при импорте обработанных данных.
3. При удалении единицы данных необходимо проверять хранимые неизвестные блоки данных ЦД на предмет связующих идентификаторов, совпадающих с идентификатором удаляемой единицы данных, для рекурсивного удаления дополняющих единиц данных, содержащих эти идентификаторы.

Исследование показало, что оптимальным языком разметки для монофайлового хранения цифрового двойника при использовании модульной структуры данных с поддержкой неизвестных блоков является xml, так как он подразумевает выполнение 2-го требования, является человекочитаемым, открывается при помощи стандартных программ большинства операционных систем, поддерживает атрибуцию и автоматическую десериализацию и сериализацию.

Рекомендуется избегать ситуаций, в которых над одним ЦД выполняют работы акторы с пересекающимися компетенциями, во избежание конфликтов данных. Если такие ситуации имеют место, то каждая единица данных должна иметь поле с датой последних изменений для разрешения конфликтов изменения данных, и глобально уникальный идентификатор для разрешения конфликта при одновременном добавлении на один идентификатор новых единиц данных разными акторами.

В качестве GUID можно использовать хэш-ключ, сгенерированный с использованием алгоритма хеширования SHA-3 на основании Unix-времени формирования единицы данных, внутреннего идентификатора в СВД и индивидуального ключа самой СВД, осуществляющей формирование этой единицы данных. Подобный подход обеспечит высокую степень уникальности идентификатора единицы данных, в том числе при массовом заполнении справочников.

Структуру описательных параметров единицы данных рекомендуется выстраивать с позиции максимальной дифференциации по смыслу (форма, тип, материал, стандарт, размер и т.п.)

#### Термины

**Цифровой двойник (ЦД):** цифровая копия конкретного физического объекта, которая отражает структуру, производительность, техническое состояние и характер рабочей миссии физического объекта, включая такие параметры, как, например, пройденные километры и возникшие неисправности.

**Система взаимодействия с данными (СВД):** программное обеспечение, осуществляющее чтение, анализ и изменение данных.

**Информационно-неоднородная система (ИНС)** в контексте работы с ЦД – это система, отдельные акторы которой не имеют единого информационного пространства и работают каждый со своими базами данных и системами обработки информации.

**Блок данных:** данные, сгруппированные по функциональному признаку, располагающиеся на верхнем уровне структуры данных цифрового двойника.

**Функциональный признак:** признак, позволяющий провести группировку и сепарацию данных по их функциональному смыслу (например, «габариты», «физические свойства», «финансовые параметры» и т.п.)

**Неизвестный блок данных (контекст: импорт цифрового двойника в систему):** блок данных, не обрабатываемый ни одним функциональным модулем СВД и, как следствие, обладающий сигнатурой, незнакомой данной СВД.

**Известный блок данных (контекст: импорт цифрового двойника в систему):** блок данных верхнего уровня, обрабатываемый одним или несколькими функциональными модулями СВД и, как следствие, обладающий сигнатурой, знакомой данной СВД.

**Функциональный модуль:** часть СВД, осуществляющая определенный набор функций, в рамках которых происходит анализ и изменение данных цифрового двойника определенным образом.

**Единица данных:** блок данных в структуре ЦД, обладающий личным идентификатором, описывающий отдельно взятый элемент. Примеры единиц данных:

- сам ЦД, обладающий личным идентификатором
- один из множества элементов списка, обладающий личным идентификатором

**Связующий идентификатор:** идентификатор, осуществляющий ассоциативную связь дополняющей единицы данных с единицей данных, находящейся в другом блоке данных.

Пример.

Блок данных участков трубопровода, содержащий перечень единиц данных, каждая из которых описывает отдельную трубу. Каждая единица данных имеет идентификатор.

Блок данных технологических параметров трубопровода, содержащий данные по допустимым, рабочим и текущим давлениям в каждой трубе. Каждая единица данных связывается с блоком данных участков трубопровода через связующие идентификаторы, эквивалентные идентификаторам соответствующих труб.

Обработывая ЦД таким образом, сперва получаем перечень труб из блока данных участков трубопроводов и заполняем параметры данными из этого блока. Затем получаем данные из блока данных технологических параметров и дополняем данные труб соответствующими по связующим идентификаторам параметрами.

## СПИСОК ЛИТЕРАТУРЫ

1. ГОСТ Р 57700.37–2021 Компьютерные модели и моделирование. Цифровые двойники изделий. Общие положения
2. Прохоров А., Лысачев М. «Цифровой двойник. Анализ, тренды, мировой опыт. Издание первое, исправленное и дополненное» 2020г.