

(19)



**Евразийское  
патентное  
ведомство**

(21) **202490600**

(13) **A1**

**(12) ОПИСАНИЕ ИЗОБРЕТЕНИЯ К ЕВРАЗИЙСКОЙ ЗАЯВКЕ**

(43) Дата публикации заявки  
**2024.05.22**

(22) Дата подачи заявки  
**2021.09.30**

(51) Int. Cl. **G06F 16/27** (2019.01)  
**G06F 16/23** (2019.01)  
**G06F 16/2453** (2019.01)  
**G06F 16/2455** (2019.01)  
**G06Q 10/06** (2012.01)  
**G06Q 10/10** (2012.01)  
**E21B 44/00** (2006.01)

**(54) СПОСОБЫ И СИСТЕМЫ ДЛЯ ОПТИМИЗАЦИИ РАСПРЕДЕЛЕННЫХ БАЗ ДАННЫХ И ДОСТУПА К СЛОЖНЫМ СТРУКТУРАМ ДАННЫХ ДЛЯ ИНТЕГРИРОВАННЫХ ПЛАТФОРМ ДЛЯ КОНЕЧНЫХ ПОЛЬЗОВАТЕЛЕЙ**

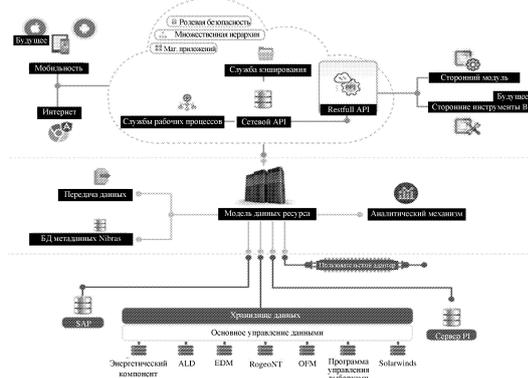
(86) **PCT/IB2021/058996**  
(87) **WO 2023/052821 2023.04.06**

(71) Заявитель:  
**ИННОВЕЙШН ТЕХНИКАЛ  
СОЛЮШН ЭсЭйОуСи  
(ИННОВАТЕК) (ОМ)**

(72) Изобретатель:  
**Мооса Рияд, Серваес Вим (ОМ),  
Кумар Ашок (ИН), Радхакришнан  
Вижайянагар, Рафик Абдул Рахман  
(ОМ)**

(74) Представитель:  
**Кузнецова С.А. (RU)**

(57) Настоящее изобретение относится к области получения доступа в цифровом формате к распределенным "большим данным" из различных ресурсов, имеющих сложные и динамические физические взаимосвязи, их моделирования и оптимизации. Настоящее изобретение также относится к системе и способу поддержки версии истины из множества источников данных и для множества сфер. Оно помогает организациям, которые владеют различными физическими ресурсами, которые используются многими группами пользователей и в течение некоторого периода времени, оптимизировать операции, осуществлять надзор и управлять сложными бизнес-процессами, связанными с их ценными ресурсами и оборудованием. Изобретение выполнено с открытой архитектурой для соединения с различными корпоративными базами данных, комбинирующими данные в реальном времени и реляционные данные. Система использует контроль по принципу исключений, сложные запросы и бескодовые способы для автоматического определения отклонений рабочих характеристик ресурса от оптимальных условий и пометки их относительно правильных пользователей в правильное время и правильным образом. Настоящее изобретение также обеспечивает возможность отслеживания состояния и рабочих характеристик оборудования и технических средств при помощи представления их полученных данных посредством интуитивных сводных панелей и автоматизированных рабочих процессов. В дополнение, мобильная версия настоящего изобретения разработана, чтобы помочь пользователям выполнять ежедневную работу, а также осуществлять управление и поддержку оборудования и ресурсов более эффективно.



**A1**

**202490600**

**202490600**

**A1**

**СПОСОБЫ И СИСТЕМЫ ДЛЯ ОПТИМИЗАЦИИ РАСПРЕДЕЛЕННЫХ БАЗ  
ДАнных И ДОСТУПА К СЛОЖНЫМ СТРУКТУРАМ ДАнных ДЛЯ  
ИНТЕГРИРОВАННЫХ ПЛАТФОРМ ДЛЯ КОНЕЧНЫХ ПОЛЬЗОВАТЕЛЕЙ**

**ПРЕДПОСЫЛКИ СОЗДАНИЯ ИЗОБРЕТЕНИЯ**

**ОБЛАСТЬ ТЕХНИКИ**

[001] Данное изобретение относится к области получения доступа к распределенным данным относительно различных физических ресурсов, их моделирования и оптимизации. Настоящее изобретение также относится к системе и способу поддержки и оптимизации доступа к системам распределенных данных, функционирующим при наличии множества географически распределенных физических ресурсов. Настоящее изобретение относится к системе и способу, которые поддерживают единую версию истины из множества ИТ систем и для множества сфер.

**[002] ОПИСАНИЕ ИЗВЕСТНОГО УРОВНЯ ТЕХНИКИ**

[003] Тогда как предшествующий уровень техники предусматривает многочисленные системы управления базами данных, ни одна из существующих систем не может интегрировать несколько функциональных средств в один цифровой двойник групп ресурсов. Такие характерные особенности, как управление базами данных реального времени и источниками связанных данных одновременно, чтобы обеспечить аналитическую информацию касательно управления оборудованием, которое работает в таких отраслях промышленности как нефтяная и газовая, не были обнаружены в предшествующем уровне техники. Присутствует ряд существующих решений по планированию ресурсов предприятия (ERP), которые могут делать это на административном уровне, при этом финансы, цепочка поставок, трудовые ресурсы и другие подобные сферы могут быть интегрированы; однако, интеграция технических ресурсов с использованием функциональных средств, заявленных в этом патенте, остается незатронутой. В основном это связано с высокой технической сложностью создания слоя абстракции данных, который может управлять выводом данных из множества

специализированного программного обеспечения и управлять этими «большими данными» таким способом, который позволяет пользователям иметь надежный и целостный обзор различных компонентов их производственных систем. Кроме того, при достижении этой интеграции посредством общепринятых способов это потребовало бы действий по кодированию для интеграции множества узлов слоев ИТ инфраструктуры их организаций, тогда как настоящие изобретения позволяют ответственным за бизнес-процессы и экспертам в предметной области создавать новые встроенные функциональные возможности в рамках платформы с использованием бескодовых интерфейсов.

[004] Многие решения предлагают создать статические иерархии различных организационных ресурсов, которые работают посредством «теговой разметки» данных из других существующих источников данных, программного обеспечения или баз данных. Патент США № 10198159B2 раскрывает систему интеграции данных, которая получает доступ к базе данных временных рядов и реляционной базе данных ресурсов. Реляционная база данных ресурсов состоит из множества узлов ресурсов, относящихся к иерархической структуре. Каждый узел ресурсов может представлять физический ресурс в промышленном применении. Система интеграции данных отображает графическое представление иерархической структуры реляционной базы данных ресурсов, получает пользовательский ввод, осуществляющий выбор узла ресурсов из иерархической структуры, и отображает шаблон иерархического поиска на основе выбранного узла ресурсов. Другие ссылки на предшествующий уровень техники представляют собой именно патенты США № 7627611B2 и № 10360217B2, в которых раскрыты способ дублирования распределенных баз данных и реализуемый компьютером способ интеграции данных соответственно.

[005] Способы и системы, описанные в предшествующем уровне техники, могут работать для небольших и управляемых источников данных, однако управление непрерывным потоком миллионов точек данных в реальном времени, т. е. «большими данными», всегда будет представлять сложную задачу. В дополнение, в таких отраслях промышленности как нефтяная и газовая важно понимать сложную взаимосвязь, которая существует между определенными наборами данных, и такие традиционные методологии, как представлено в предшествующем уровне техники, и существующие промышленные решения являются

недостаточными при управлении такими иерархиями и сложными взаимосвязями. Чтобы решить эту задачу, рассмотрим такую систему, в которой присутствует две скважины (201-А и 202-А), которые соединены посредством одного трубопровода оборудованием (301-А), и присутствует датчик давления для считывания данных с указанного трубопровода. Посредством использования скважин, дающих смесь продукции из двух или более горизонтов, в традиционных системах две скважины соединены с одним и тем же трубопроводом. Выходное значение датчика давления трубопровода имеет цифровой «тег», который управляется любой сторонней системой. Тег затем необходимо вручную добавлять дважды под каждой скважиной в традиционной статической иерархии. Проблемой такого подхода является то, что он обеспечивает пользователей неполной/неточной структурой цифрового двойника. При этом способе, если возникает проблема со скважиной 202-А, что влияет на данные трубопровода, и пользователю визуально представляются данные для скважины 201-А, пользователь не будет иметь полного понимания того, что происходит с трубопроводом, и будет не в состоянии сделать заключение о возможной первопричине. С другой стороны, настоящее изобретение может создавать эти соединения и обеспечивать пользователей лучшим обзором взаимосвязей между множеством точек данных. Кроме того, инструменты и источники данных для отслеживания двух скважин могут быть различными и могут не быть интегрированы. Это очень простой пример, но он становится намного более сложным, когда доходит до генерирования динамических иерархий и динамических тегов, которые становятся основанием для других применений, например генерирования динамических предельных диаграмм, чтобы показать ограничения производственной системы, которой необходимо учитывать такую сложность для получения надежных выходных данных.

[006] Хотя ранее упомянутый предшествующий уровень техники решает некоторые задачи, тем не менее, существует необходимость в решении проблемы распределения информации, как объяснено ранее. Существует необходимость в предоставлении интегрированного решения, которое предлагает один полезный цифровой двойник групп ресурсов, который может оптимизировать и использовать вводы «больших данных», которые генерируются от различных источников распределенных данных, для генерирования исчерпывающих и оптимизированных результатов, которые могут быть централизованы для множества сфер с точки зрения достижения единой версии истины.

Известно, что данные генерируемые от нефтяных и газовых скважин и оборудования, распределены по организационным базам данных и относятся к разным источникам программного обеспечения и вовлеченным сторонам, что также представляет другую сложную задачу; пользователям необходимо быстро и эффективно получать доступ к данным без излишнего дублирования данных при помощи импорта посредством традиционных способов, таких как процессы извлечения, преобразования, загрузки (ETL), и вместо этого при помощи виртуализации данных посредством концептуализации уникального слоя абстракции данных с целью решения этих сложных задач.

#### [007] СУЩНОСТЬ ИЗОБРЕТЕНИЯ

[008] Настоящее изобретение относится к цифровой платформе, выполненной чтобы помочь компаниям в различных секторах промышленности оптимально управлять их физическими ресурсами. Оно помогает таким организациям оптимизировать операционные возможности и расширять встроенные функциональные возможности управления ресурсами. Во-первых, это применимо когда ресурсы подобных типов необходимо отслеживать как однородные группы или управлять ими соответственно, хотя они являются изначально неоднородными вследствие их различных жизненных циклов, параметров, систем, конфигураций, связанных процессов, поставщиков или рабочих условий. Во-вторых, существуют сценарии, в которых множество сфер или организационных отделов могут получать пользу от централизованной системы обеспечения совместной работы и единой версии истины в их операционных данных. В-третьих, для сценариев, в которых соединения, связи и сети производственной системы между узлами ресурсов являются весьма сложными и обеспечивают переменное участие в других узлах в рамках общей системы, и узлы принадлежат ко множеству иерархий. Другой пример сценария предусматривает следующее, когда плановые и внеплановые процессы, уведомления, исключения и наглядное управление для одного ресурса распределены по множеству IT систем и необходимо сортировать, анализировать и определять приоритетность комплексно.

[009] Настоящее изобретение создано с помощью открытой архитектуры, которая выполнена с возможностью соединения с различными корпоративными реляционными базами данных и базами данных реального времени (посредством виртуализации). К

комплексным данным затем могут получать доступ встроенная и централизованная аналитическая машина с контролем по принципу исключений (EBS), механизм управления технологическим процессом и набор инструментов и систем визуального оповещения, сконфигурированных посредством бескодовой среды.

[0010] Система использует контроль по принципу исключений (EBS) для автоматического обнаружения аварийных отклонений параметров ресурсов от оптимальных условий и пометки их относительно правильных пользователей в правильное время и правильным образом. Для множества исключений EBS, инициируемых изнутри или при помощи внешних источников, определяется приоритетность для изоляции одного наиболее значимого исключения, чтобы избежать множества исключений для одного ресурса/элемента, чтобы избежать доведения излишних исключений до пользователей.

[0011] Изобретение использует механизм управления технологическим процессом для управления бизнес-процессами, которые могут быть связаны с управлением исключениями, выполняющий управляемые плановые задачи или внеплановые специализированные управляемые процессы, связанные с человеком. Технологические процессы также могут быть запущены для направления определенных пользователей при внесении изменений в модель данных без необходимости ввода от IT специалистов посредством бескодовых функциональных возможностей, что включает, например, добавление новых ресурсов к цифровому двойнику.

[0012] Бескодовые функциональные возможности обеспечивают внесение изменений определенными корпоративными пользователями в модель данных и отображение запросов и шаблонов, иерархий, атрибутов анализа контроля по принципу исключений, конфигураций визуальных инструментов, управления статическими и динамическими рабочими диапазонами ресурсов, конфигураций технологических процессов, отображение дерева ошибок и роли пользователя с контролем по принципу исключений.

[0013] Функциональные средства этой платформы являются возможными только при применении способов, представленных в контексте этого изобретения, поскольку представленная формула изобретения представляет собой новые способы, которые не

были ранее применены в таких целях как цифровая платформа, кратко изложенная в настоящем документе.

[0014] Настоящее изобретение обеспечивает возможность отслеживания состояния и рабочих характеристик оборудования, и технических средств при помощи представления их полученных данных посредством интуитивных сводных панелей и отчетов. В дополнение, одним из ключевых признаков настоящего изобретения является «промышленная мобильность». Мобильная версия настоящего изобретения разработана, чтобы помочь инженерам и рабочему персоналу выполнять свою работу, а также осуществлять управление и поддержку оборудования и ресурсов более эффективно. Например, это изобретение может быть использовано для предшествующих операций в отношении нефти и газа, таких как разведка, инженерно-технические работы, финансовые операции, а также другой последующей деятельности. Посредством использования этого решения EBS сотни нефтяных и газовых скважин можно отслеживать одновременно за короткий период (т. е. несколько минут), что приводит к значительной экономии на операциях, связанных с затратами и ресурсами для осуществления надзора.

[0015] Настоящее изобретение также направлено на установление значения для персонала и пользователей, которые работают в полевых условиях, вдали от офиса компании, и которые используют это изобретение на ежедневной основе, таким образом, расширяется база пользователей без ограничения высшим руководством или специалистами. Посредством применения этой методологии можно следовать более целостному подходу, что приводит к значительной адаптации пользователя в отношении организаций. Реальная польза приходит от наличия гибкой платформы, которая обеспечивает возможности для обеспечения интеграции людей, данных, процессов и технологии для достижения коммерческих целей.

#### КРАТКОЕ ОПИСАНИЕ ГРАФИЧЕСКИХ МАТЕРИАЛОВ

[0016] Изобретение будет описано со ссылкой на приложенные графические материалы, которые иллюстрируют предпочтительный вариант осуществления настоящего изобретения без ограничения объемом этого изобретения, и на которых:

[0017] на фиг. 01 представлены визуальные представления источника данных, модель данных ресурса (текущее изобретение) и соответствующие пользовательские интерфейсы. Модель данных ресурса (ADM) представляет собой цифровой двойник физических ресурсов, который извлекает данные и выполняет их обратную запись в отношении различных корпоративных баз данных, которые были созданы, чтобы служить конкретным целям. Путем интеграции корпоративных баз данных в ADM может быть выполнен тщательный анализ в отношении цифрового двойника ресурса посредством разных инструментов, например: (a) контроль по принципу исключений; (b) пользовательские рабочие процессы и управление бизнес-процессом; и (c) сводные панели и отчеты. Данные также выполнены с возможностью доступа к другим сторонним системам посредством «интерфейса программирования приложений для передачи состояния представления» (RESTful API). Платформа может быть выполнена с возможностью обратной записи и считывания сторонних узлов или исходных баз данных для обеспечения поддержания единой версии истины в области ИТ организации, что было включено в интегрированную модель данных ресурса.

[0018] На фиг. 02 показан пример традиционной модели данных ресурса (теговая разметка), которая используется в некоторых ссылках предшествующего уровня техники, где две скважины (201-А и 202-А) соединены посредством одного трубопровода с оборудованием (301-А). Датчик давления установлен на считывание и сбор данных на указанном трубопроводе. В традиционной иерархии показания датчика давления трубопровода могут быть дочерним элементом относительно родительского элемента, например «скважина 201-А или 202-А», однако при индивидуальном просмотре свойств элемента пользователи могут не быть осведомлены, что на давление трубопровода также влияют колебания давлений из другой скважины, так как они не «соединены». Это представляет собой сложные задачи при создании функциональных возможностей платформ, которые могут предусматривать соображения в отношении сложных многонаправленных взаимосвязей, которые существуют между физическими ресурсами для анализа, управления технологическим процессом и визуального оповещения. На графическом материале (b) продемонстрировано как эту сложную задачу можно решить посредством встраивания функциональных возможностей динамической теговой разметки в платформу, так что цифровое представление сложных взаимосвязей физического ресурса

может быть представлено точно. Между каждым ресурсом может присутствовать поток жидкостей или газов, которые непрерывно участвуют в технологическом процессе, и при этом теперь возможно обеспечить пользователей моделью данных, необходимой для понимания того, как эти разные потоки меняются в физически соединенных узлах ресурсов в производственной системе согласно моделируемой сложной взаимосвязи.

[0019] На фиг. 03 (А и В) показаны система и способ оптимизации рабочих характеристик запроса при осуществлении доступа к распределенным и большим наборам данных. При добавлении данных для функциональной возможности платформы запрос в виде одного массива данных выполняется в отношении слоя интеграции данных, который консолидирует параметры каждого источника данных, выбирает данные и выполняет их потоковую передачу назад к потребителю данных по мере того, как это становится доступным посредством источника данных. Этот способ обеспечивает потоковую передачу множества асинхронных событий потребителю данных быстрее, чем общепринятый способ, при этом устраняются задержки, когда потребитель данных извлекает источник данных (источники данных) с самым медленным откликом. На фиг. 03. В показан процесс консолидации запроса для сложных иерархий.

[0020] На фиг. 04 (А и В) показаны система и способ для оптимизации доступа к распределенным данным. Тот же самый параметр может быть извлечен из (и записан в них) разных слоев в архитектуре информационных технологий организации в зависимости от контекста и цели. Очень часто присутствуют задействованные системы хранения данных и потенциально еще больше слоев кэширования для наличия лучших рабочих характеристик извлечения данных. Однако требования к обновлению данных могут создавать некоторую сложность. Если обновление выполняется в том же слое, где данные были считаны (в случаях отличных от исходной базы данных), тогда данные не отражаются немедленно в первоисточнике, что вызывает несоответствия в данных в разных узлах, что может потенциально приводить к дезорганизации. Если данные обновляются в первоисточнике, будет присутствовать задержка для отражения изменения в разных слоях, поскольку эти разные слои не обязательно имеют возможность принудительного обновления кэша по запросу. Следовательно, временно считывание

может быть выполнено с возможностью осуществления из исходной базы данных до отражения данных в последующих слоях.

[0021] На фиг. 05 (А и В) показаны система и способ для оптимизации моделирования данных физического ресурса. Ресурсы развиваются и переходят на разные фазы, что дает им изменение параметров в течение разных периодов времени. В дополнение, некоторые из данных могут быть доступны для полного жизненного цикла ресурса, но источник данных или логика анализа могут изменяться с течением времени, что может приводить к техническим ограничениям для функциональных возможностей, например когда выполняется долгосрочный хронологический анализ ресурса.

[0022] На фиг. 06 показаны система и способ для оптимизации моделирования данных физического ресурса. Цифровой двойник получают посредством шаблонов данных для обеспечения стандартизации и эффективной конфигурации. Цифровой двойник состоит из корневого элемента и может иметь множество слоев дочерних шаблонов, которые являются или обязательными, или необязательными и могут быть динамичными по своему количеству. Правила шаблонов будут определять динамические характеристики образцов шаблонов. Любые запросы из слоя абстракции будут динамически возвращать параметры согласно сконфигурированной модели данных.

[0023] На фиг. 07 представлены система и способ для поддержки одного источника данных для множества сфер в отношении достижения золотой записи. В многопрофильной организации одни и те же ресурсы отслеживаются в рамках разных контекстов и целей и посредством разных систем и баз данных. Чтобы служить этим различным целям, большую часть времени существуют разные иерархии, требуемые для обеспечения правильного анализа для сферы. У этого могут быть свои последствия в отношении интеграции данных для сохранения ресурса только однажды в базе данных для обеспечения того, что единая версия данных (истины) предоставляется другой аудиторией. Для обеспечения этого, такая же запись может быть связана тегом с другими иерархиями. Альтернативно, в случаях, в которых множество записей могут существовать для общей точки данных, существуют механизмы для обеспечения того, что золотая запись может быть выделена без необходимости удалять другие иногда целесообразные записи из модели данных.

[0024] На фиг. 08 (А и В) показан полный набор инструментов для обеспечения других бескодовых функциональных возможностей, чтобы охватить весь цикл создания приложения. Слой абстракции данных повышает уровень слоев абстракции сложных и кодированных данных до конечных пользователей, чтобы они могли конфигурировать виртуализацию данных соответственно без необходимости дублирования данных в базе данных в бескодовой среде, что было бы сделано в рамках существующего подхода. Слой абстракции данных можно виртуализировать и соединять с внешней базой данных для бескодовых сред без необходимости импорта или хранения данных в рамках платформы в первую очередь. Пользователям доступен интерфейс для конфигурирования этих соединений и запросов/управления ими, что приводит в результате к полной комплексной бескодовой среде/платформе.

[0025] На фиг. 09 показан пример производственных мощностей и отклонений агрегированной производственной системы, учитывая сложные взаимосвязи ресурсов, и продемонстрировано сильное влияние, которое точный цифровой двойник может оказывать посредством предоставления обзора всех возможных источников отклонений в центральном положении вне зависимости от используемого программного обеспечения, типа ресурса или источника данных (команда / сфера / область / группа). Фильтры также применяются, и данные, к которым получают доступ, ограничены ролями с соответствующими идентификационными данными.

[0026] На фиг. 10 показан пример визуального представления простого доступа ко всей требуемой информации для отслеживания группы рабочих характеристик ресурсов и управления ею. Представленные графики предусматривают согласование добычи, нагнетания, отклонений и поверхностного или подповерхностного рабочего диапазона в одном месте.

[0027] На фиг. 11 показан пример визуального представления рабочих характеристик группы ресурсов (т. е. скважин) для выбранной функциональной группы. Это способствует увеличению эффективности управления ресурсами и реализации альтернативной стоимости. Принципы контроля по принципу исключений и экономии были использованы как выбранные подходы для сведения к минимуму затраченного времени комплексных процессов при управлении ресурсами и их отслеживании.

[0028] На фиг. 12 показан пример передачи данных в реальном времени от датчиков на пользовательский интерфейс (т. е. рабочий стол) для реализации обнаружения неисправностей, связанного с проблемами качества данных, например обеспечение линейности данных, выход за пределы диапазона, прерванные передачи данных, для идентификации наиболее вероятной первопричины. Настоящее изобретение способствует агрегированию и анализу данных для примеров, таких как электропитание на объекте, конфигурация удаленного терминального устройства (RTU), телекоммуникационные башни, конфигурации серверов связи на открытой платформе (OPC) и конфигурации «тегов» стороннего программного обеспечения. Такая характерная особенность способствует изоляции одной неисправности на физический элемент в случаях, когда разными системами могло быть вызвано множество исключений, несмотря на присутствие одной первопричины (т. е. нарушения связи, как продемонстрировано на фиг. 12).

[0029] На фиг. 13 показан пример производственных мощностей интегрированной производственной системы относительно фактического производства для анализа оптимизации всей производственной системы. Это приложение показывает диаграммы предельных значений согласно ограничениям производственных мощностей нефти, воды или газа. Изменения обводненности и коэффициентов содержания газа в нефти (GOR) можно имитировать, и долгосрочные тенденции ограничений можно генерировать для отслеживания улучшений в отношении интегрированной производственной мощности системы.

[0030] На фиг. 14 предоставлен пример визуального представления данных отклонений для добывающих скважин группы ресурсов с течением времени, что помогает уменьшить время, требуемое для анализа пользователями хронологии данных отклонений, чтобы идентифицировать приоритетное направление для анализов рабочих характеристик (что затем приводит в результате к идентификации возможностей уменьшения отклонения). Это также способствует тому, что данные отклонений становятся видимыми и доступными для команд, связанных со всей производственной системой. Влияние отклонения одного физического ресурса может быть зарегистрировано как «групповое» отклонение, что добавляет соображение в отношении эстафетного эффекта, который отклонение одного

ресурса оказывает на другие связанные элементы на основе сложных взаимосвязей, которые определяет модель данных платформы.

[0031] На фиг. 15 (А и В) предоставлен пример отслеживания нефтепромыслового оборудования с использованием настоящего изобретения. Насосы, компрессоры, резервуары и сепараторы отслеживаются посредством исключения таким образом, чтобы можно было измерить состояние ресурса. Исключения генерируются, если обнаруживаются сдвиги рабочих диапазонов с точки зрения производительности оборудования, состояния контроллера и состояния подсистемы, или когда сторонние специализированные системы инициируют исключения. Когда платформа и другие сторонние системы инициируют множество «активных» исключений, для исключений можно определить приоритетность, и можно сконфигурировать анализ для выделения одного наиболее соответствующего или фактического исключения первопричины для доведения до пользователя. Диагностический график затем доступен для предоставления обзора соответствующих данных, необходимых оценки критических данных в случаях, когда анализ платформы потребует интерпретации человеком или принятия решения в отношении требуемого действия.

[0032] На фиг. 16 предоставлен пример возможностей динамического оповещения для анализа в реальном времени данных, связанных с химическими процессами, в качестве примера. Это способствует улучшению целостности отслеживания. Данные из множества разрозненных источников могут быть перемещены при необходимости пользователями для выполнения комплексного анализа. Вводы данных по умолчанию могут быть сконфигурированы таким образом, чтобы автоматически заполняться в отдельном приложении платформы.

[0033] На фиг. 17 предоставлен пример автономии и способности обновлять и регулировать конфигурацию программного обеспечения конечного пользователя с использованием обновления модели данных в качестве примера для обеспечения точности модели. При широкомасштабном развертывании существует необходимость в обеспечении этих инструментов для группы операторов ресурса, поскольку команды поддержки информационных технологий не обязательно вовлечены в инструментальное оснащение ресурсов.

[0034] На фиг. 18 (А и В) обеспечены снимки экрана, которые обеспечивают обзор объема параметров конфигурации запроса согласно настоящему изобретению, которые доступны конечному пользователю. Пользователи могут использовать интерфейс для выбора из многих предварительно установленных соединений, доступных в строке отображения данных. На фиг. 25 В показан визуальный отчет, который был создан с использованием бескодового инструментария платформы.

## **ПОДРОБНОЕ ОПИСАНИЕ ИЗОБРЕТЕНИЯ**

[0035] Ссылка на приложенные фигуры выполнена для обеспечения более всеохватывающего понимания различных способов и устройств, раскрытых в настоящем документе. Дополнительные графические материалы считаются графическими представлениями, выполненными с целью объяснения и для обеспечения только примеров и демонстраций, и, следовательно, не предназначены ни для идентификации габаритов, размеров, ни определения/ограничения объема раскрытого изобретения.

[0036] Конкретные термины были использованы с целью предоставления ясного описания этого изобретения, однако эти термины не предназначены ни для определения, ни ограничения объема изобретения, используются для ссылки на конкретную структуру различных вариантов осуществления.

[0037] Варианты осуществления настоящего изобретения могут быть реализованы в интегрированной системе баз данных, такой как показанная на фиг. 01. Модель данных ресурса представляет собой цифровой двойник физических ресурсов, извлекающий данные из других корпоративных баз данных, которые был создан, чтобы служить конкретным целям. Путем комбинирования корпоративных баз данных в ADM может быть выполнен тщательный анализ в отношении цифрового двойника ресурса посредством разных инструментов, например: (а) контроль по принципу исключений; (b) рабочие процессы и управление бизнес-процессом; и (с) сводные панели и отчеты. Данные также представлены другим пользователям посредством «интерфейса программирования приложений для передачи состояния представления» (RESTful API). ADM стандартизирует названия параметров и модулей для случаев, в которых информация собирается от множества источников. Она также поддерживает стандартные технологии

системы управления реляционными базами данных (RDBMS), а также больше интерфейсов, являющихся чьей-то собственностью, для данных в реальном времени, SAP для финансовых/эксплуатационных данных и других служб RESTful от сторонних специализированных систем.

[0038] Для большого количества записей тот же параметр может происходить из разных баз данных, характеризующихся разными строками соединения; в таких случаях поступления слоя абстракции данных могут быть сконфигурированы таким образом, чтобы возвращать их к значению общего параметра, так что любой потребитель данных может обрабатывать данные систематически без влияния существующей разнородной корпоративной инфраструктуры. Следовательно, названия параметров и модулей в исходных базах данных стандартизируются в рамках платформы. В свою очередь, это делает возможным просмотр агрегированных производственных данных, имеющих разные модули (из множества источников данных), в одном интегрированном средстве визуализации. Единицы измерений могут быть установлены без необходимости внесения изменений в конфигурации запросов. Запросы относительно серверной части будут динамически построены на основе запроса данных. Разные единицы измерений (UOM) из разных источников данных преобразуются в стандартизированное условное обозначение UOM, как указано конфигурацией потребителя данных. Цифровые двойники физического ресурса основаны на шаблоне для обеспечения возможности простого дублирования огромного количества конфигураций запросов в контексте управления «большими данными». Элементы, которые конфигурируются посредством разных шаблонов, в течение времени будут иметь разные версии, так что извлечение статистических данных будет автоматически переключаться между этими разными версиями и разными конфигурациями шаблонов. Это обеспечивает поддержку управления жизненным циклом ресурсов. Например, нефтяная скважина может изначально быть классифицирована в соответствии с иерархиями разведки и следовать модели шаблона данных, которая соответствует этим требованиям на момент времени. Как только ресурс передается в эксплуатацию и становится добывающей скважиной, тот же шаблон ресурса будет изменен для отражения его добывающего состояния. Наборы данных, генерируемые и собираемые с такой скважины, отличаются существенно между этими двумя фазами. Кроме того, во время фазы эксплуатации она может изначально работать как свободно

фонтанирующая скважина, но затем быть переоборудована под искусственный подъем различных типов. На протяжении всего этого жизненного цикла скважины остаются такими же, как ID ресурса и элемента, но собранные параметры, версии и данные значительно отличаются на протяжении ее жизненного цикла и могут требовать соответствующих статистических данных в некоторых контекстах.

[0039] В другом предпочтительном варианте осуществления настоящего изобретения при добавлении данных для функции платформы запрос в виде одного массива данных выполняется в отношении слоя абстракции данных платформы, который консолидирует параметры каждого источника данных, выбирает данные и выполняет их потоковую передачу назад к потребителю данных по мере того, как это становится доступным посредством источника данных. Этот способ обеспечивает потоковую передачу множества асинхронных событий потребителю данных быстрее, чем общепринятый способ, при этом устраняются задержки, когда потребитель данных извлекает источник данных (источники данных) с самым медленным откликом. Когда запрос отправлен на сервер (содержащий множество точек данных), этот запрос разделяется источником данных, и данные выводятся клиенту асинхронно. Это означает, что он может заполнять данные для одной записи, пока другие записи ожидают ответа сервера для заполнения. В свою очередь это означает, что экран пользователя не замрет, если база данных отсутствует или не функционирует, что приводит к реагирующему экрану независимо от простоев в работе или медленных баз данных. Обычно такие сложные задачи решаются использованием таких способов, как «буферизация», чтобы обеспечить возможность достаточного времени для баз данных для возврата записи, когда существует разница между скоростью, с которой данные получают, и скоростью, с которой они могут быть обработаны. Однако для предложенного способа все наоборот, при этом извлечение данных может быть отделено от скорости отклика для вывода клиенту (записи, которые могут быть заполнены немедленно, становятся доступными, тогда как записи из источников данных с более медленными скоростями отклика будут заполнены в своем собственном темпе). На фиг. 03-В на схеме показан пример процесса и способов, используемых в рамках слоя абстракции данных настоящего изобретения для обеспечения заявленных функциональных возможностей для работы.

[0040] В другом предпочтительном варианте осуществления настоящего изобретения присутствует способ виртуализации данных, который является подходом к управлению базой данных, который при правильной реализации должен быть уникальным и единым источником истины. Виртуализация данных может эффективно объединять данные в хранилищах данных, витринах данных, озерах данных и сторонних источниках данных без необходимости создания полностью новой интегрированной платформы физических данных. Существующая инфраструктура данных может продолжать выполнять свои основные функции, тогда как слой виртуализации данных эффективно использует данные из этих источников. Этот аспект виртуализации данных делает ее дополняющей в отношении всех существующих источников данных и увеличивает доступность и использование данных предприятия. Слой абстракции данных интегрирует эти способы виртуализации, чтобы сделать эти данные видимыми для конечного пользователя клиентского приложения посредством данных, которые относятся к усовершенствованным принципам управления запросом, которые применяются к слою платформы конечного пользователя и интегрируются с ним вместо общепринятого (общепринятых) слоя (слоев) управления корпоративной базой данных, где такие способы обычно используют, что делает усовершенствованные возможности управления данными на один шаг ближе к конечным пользователям. В отличие от традиционного процесса извлечения, преобразования, загрузки («ETL») данные остаются на месте, и доступ в реальном времени предоставляется к исходной системе для данных. Это уменьшает риск ошибок данных и устраняет необходимость в вычислительных ресурсах, связанных с перемещением и хранением данных, которые, возможно, никогда не будут использованы, и нет попытки навязать одну модель данных в отношении данных. Технология также поддерживает обратную запись обновлений данных транзакции на исходные системы. Для устранения различий в форматах и семантике источника и потребителя используются различные методы абстрагирования и преобразования. Виртуализация данных может также считаться альтернативой ETL и хранению данных в хранилищах данных. Виртуализация данных изначально направлена на быстрое и своевременное получение информации от множества источников без необходимости начинать большой проект в области данных со значительными ETL и хранением данных.

[0041] Другим предпочтительным вариантом осуществления настоящего изобретения является способность эффективно комбинировать данные из широкого спектра разрозненных источников данных организации. Диаграмма предельных значений является одним примером инструмента визуального оповещения, который требует учета отклонений, которые возникают, когда ресурс в сложной интегрированной производственной сети имеет перерыв в производстве в отношении скважин, модулей, поездов или платформ, вызванный аварийной ситуацией. Случаи отклонений обычно хранятся в базе данных по распределению углеводородов и используются для ведения записей событий отклонений. Настоящее изобретение позволяет пользователям управлять этими точками данных, визуализировать, редактировать, считывать и/или записывать их, когда они влияют на другие приложения/базы данных программного обеспечения в рамках настоящего изобретения или другое стороннее программное обеспечение. Возможности по созданию цифровых двойников, которыми характеризуется настоящее изобретение, позволяют пользователям обновлять соединения (отображения данных) в одном месте и обеспечивать воздействие, дублируемое в других связанных функциональных возможностях платформы (которые выполнены так, чтобы действовать как интегрированная система управления ресурсами разных организационных сфер/существующей ИТ инфраструктуры), и автоматизировать процессы сбора данных и подтверждения. На фиг. 09 показан пример отклонения «других элементов», явно демонстрирующий мощное воздействие, которое точный цифровой двойник может иметь, посредством предоставления обзора всех возможных источников отклонений в одном центральном положении независимо от используемого программного обеспечения, типа ресурса или источника данных (команды / сферы / области / группы). Посредством этой полученной информации пользователи могут отображать воздействие события отклонения на их интегрированные производственные мощности динамически. Фильтры также применяются, и данные, которые отображаются, ограничены ролями с соответствующими идентификационными данными.

[0042] В другом предпочтительном варианте осуществления настоящего изобретения слой абстракции данных не хранит каких-либо данных, вместо этого он указывает и считывает данные со сторонних баз данных. Таким образом, когда необходимо внести корректировки, платформа позволяет пользователям вносить изменения в записи

непосредственно в источнике данных, чтобы обеспечить отображение изменений в платформе, но также для любой другой системы, потребляющей исходные данные. Хотя это может препятствовать процессу автоматического и непрерывного получения обновленных данных, тем не менее, это обеспечивает уверенность в единой версии истины. Для усиления этого принципа модель данных также раскрывается посредством множества RESTful API, так что потребление данных не ограничено пользователями только настоящего изобретения, но также для сторонних организаций и владельцев данных, и их привилегированных потребителей данных.

[0043] В предпочтительном варианте осуществления настоящего изобретения раскрыт реализуемый компьютером способ сбора, интеграции и представления данных. Способ включает предоставление соединителей хранилища данных, слоя абстракции данных, слоя интеграции данных и графического пользовательского интерфейса, при этом набор данных участвует в обмене между указанным хранилищем данных и указанным слоем абстракции данных и между указанным слоем абстракции данных и указанным слоем интеграции данных; конфигурирование собранных точек данных в цифровые двойники, представляющие физические ресурсы; комбинирование запросов в массовые запросы для уменьшения частоты запросов к интерфейсам программирования приложений сторонних систем; обеспечение представления множества асинхронных событий и потоковой передачи указанных событий в отношении потребителя данных; и отображение графического представления указанных многих асинхронных событий, использующих указанный графический пользовательский интерфейс.

[0044] Другой предпочтительный вариант осуществления настоящего изобретения состоит в том, что когда запрос конкретного параметра не получается, автоматические запросы нейтрализации неисправности являются конфигурируемыми для продолжения извлечения данных из альтернативных баз данных резервного копирования. Это поддерживает системы хранения данных, когда неожиданно происходят отключения, и, следовательно, запрос нейтрализации неисправности будет обходить хранилище данных для автоматического перенаправления на исходную базу данных. В дополнение, запросы считывания и записи могут быть выполнены с возможностью указания на различные базы данных. Например, когда система хранения данных находится между исходным

сторонним источником данных и платформой, и приложениями представленного изобретения, операции считывания могут быть выполнены в отношении системы хранения данных, тогда как операции считывания могут быть выполнены с возможностью непосредственного указания на систему сторонних источников данных.

[0045] В другом предпочтительном варианте осуществления настоящего изобретения тот же параметр может быть извлечен из (и записан в них) разных слоев в архитектуре информационных технологий в зависимости от контекста/цели. Часто существуют системы хранения данных и потенциально еще больше слоев кэширования для стабилизации рабочих характеристик системы разных корпоративных баз данных. Однако эти механизмы кэширования могут увеличивать сложность, когда записи необходимо обновлять или исправлять. Если обновление выполняется на уровне, отличном от исходного источника данных, на данные может влиять разная частота обновления механизмов кэширования, которые могут существовать между слоями и приводить к несоответствиям данных, которые могут потенциально приводить к неправильным операциям считывания запроса, поскольку эти разные слои не обязательно присутствуют для обеспечения обновления кэша по требованию. Следовательно, временно считывание должно происходить из исходной базы данных до отражения данных в верхних слоях. Например, в сферах нефти и газа результаты испытаний скважины считываются с базы данных по распределению углеводородов обычно в рамках системы хранения данных, которая может характеризоваться механизмами кэширования между слоями. Запись в хранилище данных не обязательно является обратным обновлением данных в системе исходных баз данных; следовательно, обратная запись в исходную базу данных становится необходимостью для отражения изменений. Впоследствии все отчеты, использующие эти данные, должны показывать эту обновленную информацию; однако присутствует задержка вследствие разных частот обновления. В этих случаях она может быть сконфигурирована таким образом, чтобы также перенаправлять запросы считывания в исходную базу данных.

[0046] В контексте архитектур информационных технологий, характеризующихся многими слоями/уровнями, иногда требуется записать что-то обратно в базу данных, и это может повлиять на другие параметры, которые необходимо повторно считать. Обычно

считывание происходит сверху (т. е. из хранилища данных). Но затем, если необходимо что-то записать и изменение значения влияет на другие значения, которые вычисляются в исходной базе данных, требуется запустить «обновление» других слоев для получения нового значения. Вот почему обновленное значение в хранилище данных пока не может быть извлечено, как будто присутствует задержка обновления. Это может быть устранено посредством записи и считывания непосредственно в отношении исходного приложения для того же значения или того же параметра.

[0047] В дополнение, настоящее изобретение имеет множественное соединение с разными базами данных, которые могут динамически «переключаться» на считывание/запись для разных обстоятельств и контекстов. Эта концепция применяется в контексте, в котором пользователь может конфигурировать правила динамического запроса для виртуализации данных, которые определяются операционными требованиями (более быстрые скорости обновления данных) в дополнение к функциональным средствам отказоустойчивости. Пример предыдущего функционального средства является тем, что представлено на фиг. 04, в нефтяных и газовых скважинах, что демонстрирует влияние на данные в зависимости от того, где они считываются/записываются. В некоторых изолированных сценариях, при необходимости, пользователь, желающий изменить точку данных «В-Status / Not Approved» на «В-Status / Approved» традиционно сталкивается со сложностями, когда данные не перезаписываются непосредственно в исходной базе данных. Это связано с рядом слоев, которые могут существовать между исходными данными и потребителем, что может вызывать задержки из-за логики различных систем, таких как механизмы кэширования, для управления общими рабочими характеристиками систем. Как показано на фиг. 04-А и объяснено ранее, в данном случае сторонний инструмент (испытательный инструмент для скважины) будет взаимодействовать с корпоративным хранилищем данных для записи данных в «базе данных углеводородов» (может быть любая другая база данных), которая пребывает в корпоративном хранилище данных. С этого момента база данных ЕС может синхронизировать данные со сторонним приложением, которое содержит исходный источник данных. Стороннее приложение также может иметь свой собственный механизм кэширования, который может задерживать, кроме того, весь процесс.

[0048] На фиг. 18-А показаны различные параметры операции запроса, доступные пользователям для выбора. Разница между «Write» и «WRITE TO SOURCE» продемонстрирована в предыдущем рассмотренном изображении. В зависимости от того как операции сконфигурированы на этом экране, данные будут «потреблены» программным обеспечением по-другому. В конечном итоге, назначение этих функций состоит в обеспечении того, чтобы правильное равновесие между рабочими характеристиками системы (управление «большими данными») и функциональными возможностями системы постоянно оптимизировалось до конкретных контекстов, в которых требуются данные. На фиг. 18-В проиллюстрирован пример использования параметров запроса операций вывода в контексте визуального отчета, конфигурируемого пользователем. Экран настройки конфигурации, ранее показанный на фиг. 18-А, в конечном итоге приводит к этому выводу (несмотря на другие этапы в процессе). Точное и эффективное заполнение данных для таких визуальных представлений требует тысяч и тысяч точек данных, размещенных в распределенных и многоуровневых хранилищах данных и системах, и требует чтобы эта комплексная интеграция инструментов управления данными, например это требование, была выполнена с возможностью управления без вмешательства ИТ персонала (бескодовая среда).

[0049] В предпочтительном варианте осуществления настоящего изобретения раскрыт реализуемый компьютером способ сбора, интеграции и представления данных. Потребитель данных дополнительно предусматривает: обеспечение набора запросов считывания и записи; генерирование запроса считывания для сбора данных из разрозненных организационных источников данных для предоставления анализа и отчетов; генерирование запроса записи для изменения записей, полученных посредством указанного запроса считывания, при этом такой запрос выполнен с возможностью изменения записей в базе данных исходного приложения; показ обновленной записи посредством нового запроса считывания; перенаправление указанного запроса считывания автоматически для инициирования нового запроса считывания в базу данных исходного приложения вместо хранилища данных; и возврат скорректированной и обновленной записи потребителю данных.

[0050] В предпочтительном варианте осуществления настоящего изобретения раскрыт способ управления физическими ресурсами, вовлеченный в различные фазы их жизненного цикла. В отраслях промышленности, которые предусматривают изменения и стадии в работе, физические ресурсы переходят в разные фазы, что дает им или больше, или меньше данных во время их эксплуатационных жизненных циклов. Некоторые из данных могут быть доступны для полного жизненного цикла ресурса, но другие данные могут изменяться с течением времени. Это является потенциальной проблемой, если необходим долгосрочный анализ.

[0051] Например, в нефтяной и газовой отрасли промышленности новая скважина, которая пробурена, начинается с идентификации в качестве элемента или ресурса в финансовых системах, при этом ограниченные данные записываются. Затем она переходит в эксплуатационную фазу, где она может пребывать в качестве свободно фонтанирующей добывающей скважины и затем может быть изменена для использования искусственного подъема различных типов, которые необходимо учитывать. (Каждая из этих фаз предусматривает разные источники данных и атрибуты). В конечном итоге скважина может быть преобразована в нагнетательную или наблюдательную скважину до конца ее срока службы. Во время всего этого цикла модель ресурса пометит тегом данные, связанные с физическим отверстием в земле. Используя анализ эффективности насоса в качестве примера, эти параметры могут быть обнаружены только во время периода, когда система искусственного подъема была физически применена, тогда как расчеты эффективности насоса сами по себе будут, к тому же, изменяться согласно типу искусственного подъема, который был применен. Напротив, конструктивные данные должны быть доступны во время всего жизненного цикла скважины. В другом примере учитывается долгосрочный анализ резервуара, при этом скважина может быть преобразована с добывающей в нагнетательную, и при этом построение кривой iVRR (отношения замещения пористости) потребует незаметного для пользователя комбинирования обоих наборов данных. Слой абстракции данных платформы делает возможным комбинирование производственных данных, которые доступны в течение периода времени, с данными нагнетания из другого периода, дублирующими то, что физически происходит с физической скважиной. Подобным образом, будут доступны данные для конкретного бизнес-процесса, которые актуальны только, пока процесс

активен и в рамках контекста этого процесса. Как показано на фиг. 05-В, физический ресурс всегда остается таким же независимо от сферы, но им можно динамически управлять для отражения состояния жизненного цикла ресурса и его изменяющихся источников данных, не теряя из виду статистические данные, которые принадлежали этому «фиксированному» элементу/ресурсу.

[0052] Управление исходным кодом и управление версиями, где файлы регистрируются и замещают друг друга, при этом поддерживается запись, является другим примером, в котором это изобретение может быть использовано. Обычно разные версии активны в разные периоды времени, предоставляя или разные наборы данных посредством разных каналов, или множество наборов данных из разных источников данных. В традиционном управлении версиями большую часть времени используется конкретная версия, затем или создается новая версия, или происходит откат до предыдущих версий, тогда как в настоящем изобретении версии комбинируются посредством модели данных. Когда эта модель изменяется с течением времени по каким-либо причинам (т. е. изменение датчика, изменение тега, изменение модели данных, изменение источника данных тега и т. д.), настоящее изобретение может агрегировать различные версии, чтобы создавать полные функции, такие как совокупные тенденции. Это является целесообразным, когда пользователи хотят увидеть статистические данные (т. е. периоды между сегодня и 3 годами ранее) для добавления исторического контекста, который помогает понять полный жизненный цикл ресурса. Это достигается посредством поддержания записи всех версий и посредством создания слоя абстракции данных, который может управлять этими данными в контексте других критических функциональных возможностей платформы. Например, когда платформа анализирует долгосрочные тенденции для предсказания выходов ресурса из строя, она должна иметь возможность учета изменений в версиях для платформы для возврата надежных выходных данных и исключения неправильных наборов данных. Например, состояние датчика давления физического ресурса изменялось три раза на протяжении трех лет; в первом году присутствовал датчик давления, во втором году этот датчик давления был заменен датчиком другого типа, и в третьем году была изменена конфигурация тега датчика давления в сторонней исходной базе данных для отражения изменения типа ресурса. Связь с этими тремя хронологическими событиями или изменениями в жизненных циклах иначе была бы утрачена без уникального способа

абстракции данных, который может эффективно обрабатывать функцию датчика давления в качестве фиксированного элемента, который может иметь множественные или изменяющиеся источники входных данных, подлежащие отслеживанию с течением времени.

[0053] В предпочтительном варианте осуществления настоящего изобретения раскрыт реализуемый компьютером способ сбора, интеграции и представления данных. Способ включает: запись изменений в точки данных, которые представляют множество физических ресурсов в реляционной базе данных событий; получение доступа к указанной реляционной базе данных событий для генерирования новых тегов для замен физических ресурсов; проверка наличия у новых тегов таких же функции и роли, что и у сохраненных в хранилище данных; конфигурирование указанных новых тегов посредством нового идентификатора версий атрибута, когда новые теги имеют те же функцию и роль, как и те, которые сохранены в хранилище данных, и связывание всех данных из разных идентификаторов версий атрибутов для представления полного набора данных; и конфигурирование указанных новых тегов посредством нового идентификатора атрибута, когда новые теги не имеют такие же функцию и роль, что и сохраненные в хранилище данных, и передача изменений в точки данных временных рядов для одного идентификатора атрибута.

[0054] В настоящем изобретении цифровой двойник для физических ресурсов создается посредством шаблонов данных для обеспечения стандартизации и эффективного конфигурирования. Цифровой двойник состоит из корневого элемента и может иметь множество слоев дочерних шаблонов, которые являются или обязательными, или необязательными и могут быть динамичными по своему количеству. Правила шаблонов будут определять динамические характеристики образцов шаблонов. Любой запрос из слоя абстракции будет возвращать параметры дочерних элементов динамичным образом. Создаются цифровые двойники физических ресурсов посредством этих шаблонов, и они обеспечивают применение глобальных изменений ко всем ресурсам, связанным с шаблоном. Например, выполнение изменений конфигурации шаблона для тега давления датчика в одном шаблоне может обновлять все теги давления для каждого ресурса,

который связан с шаблоном, без необходимости изменений в коде. Шаблоны могут содержать где-то от одного до сотен тысяч связанных ресурсов или элементов.

[0055] В качестве примера, компрессор является корневым элементом, но будет иметь ряд шаблонов высокоуровневого компонента, например ступень передачи и сжатия. Компрессор не может работать, если не доступен привод, поэтому это будет обязательный дочерний элемент, который имеет все разные параметры привода. Когда создается новый экземпляр компрессора, привод будет обязательным элементом, подлежащим конфигурированию. Подобным образом ступень сжатия является обязательной, но могут присутствовать компрессоры с более чем одной ступенью. Настоящее изобретение предлагает вариант комбинирования всех шаблонов (всех шаблонов для одного и того же типа оборудования, имеющего разные типы моделей) посредством точного указания того, какие компоненты (атрибуты) шаблонов являются обязательными, а какие являются необязательными, как изображено на фиг. 06. Это также обеспечивает полную гибкость для пользователей с соответствующими идентификационными данными для конфигурирования шаблонов так, чтобы они были точным повторением их рабочих условий. Со знанием того, какие компоненты (атрибуты) являются обязательными, настоящее изобретение может обеспечивать функциональные возможности намного более эффективно. Используя визуализацию в качестве примера, шаблоны компрессора могут быть выбраны для просмотра обязательных и необязательных выходных данных (если они доступны) в той же тенденции. При создании формул для анализа, используемых в контексте контроля по принципу исключений, например, создается анализ для шаблона компрессора, формула может автоматически проверять все указанные атрибуты, учитывая необязательные атрибуты.

[0056] Для создания изначального отображения данных (когда решение впервые развертывается и конфигурируется в первый раз) отображение данных займет в три раза дольше в этом сценарии, поскольку конфигурация атрибутов будет излишне воспроизведена в трех шаблонах вместо одного общего шаблона (в случаях обязательных атрибутов). Посредством конфигурирования атрибутов, которые являются общими для всех компрессоров в одном шаблоне, эффективность процесса значительно улучшается. Этот простой принцип/способ применяют для последовательного конфигурирования

приложений или функциональных возможностей в рамках настоящего изобретения на основе этого изначального отображения данных. Например, если требуется создать формулу анализа для инициирования исключения, когда тег давления датчика считывает значения, которые опускаются ниже заданного порогового значения, это потребует сконфигурировать это множество отдельных раз с использованием общепринятых способов иерархии и шаблона, тогда как тот же результат может быть достигнут посредством выполнения этого только один раз посредством методологии настоящего изобретения для управления иерархиями и шаблонами.

[0057] Также необходимо выделить, что создание визуализации ресурсов (цифрового двойника) преобразуется в результате этого динамического подхода, поскольку модель гибкого шаблона может динамически влиять на визуальный вывод на основе отдельного элемента в зависимости от компонентов (атрибутов), которые доступны для данной единицы оборудования. Цифровые двойники могут потребоваться для отражения цифровых представлений их физических свойств, которые могут изменяться для оборудования разных типов, имеющего ту же базовую функцию (компрессоры с разным количеством ступеней). Например, вместо наличия графического представления типового изображения для каждого типа компрессора компоненты компрессора создаются на основе атрибутов, доступных в рамках шаблона для специализированного элемента компрессора, и можно, следовательно, создавать изображение «цифрового двойника», которое превосходно представляет отличия между каждой единицей оборудования, которая принадлежит к одному и тому же шаблону, и как продемонстрировано на фиг. 06. Следовательно, методология настоящего изобретения обеспечивает то, что единая версия истины достигается посредством обеспечения конфигурации, при необходимости, с помощью необязательных атрибутов и фиксации необходимых атрибутов посредством обязательных атрибутов. В качестве примера ресурс А имеет компрессоры типа А, тогда как ресурс В имеет компрессоры типа В. Оба компрессора являются разными моделями и отслеживают разные наборы параметров. Однако пользователь может захотеть просмотреть ввод/вывод обоих типов агрегированных компрессоров. Для достижения этого в их отношении конкретизировано, какие вводы/выводы для обоих типов компрессоров являются «обязательными» и необходимо сконфигурировать, а какие

параметры являются необязательными и можно конфигурировать на усмотрение каждого ресурса оперативной группы.

[0058] В предпочтительном варианте осуществления настоящего изобретения раскрыт реализуемый компьютером способ сбора, интеграции и представления данных. Способ предусматривает цифровые двойники, которые имеют один функциональный элемент и несколько переменных дочерних элементов, которые сконфигурированы в рамках шаблонов цифровых двойников, при этом один шаблон создается для всех элементов, которые имеют такую же функцию с необязательными атрибутами, где это применимо, конфигурация атрибута анализа выполняется по меньшей мере один раз без влияния на нее других необязательных атрибутов, и разные результаты, основанные на необязательных атрибутах, возвращаются посредством использования булевой логики.

[0059] Другим предпочтительным вариантом осуществления настоящего изобретения является интегрированная иерархия ресурса. На фиг. 02 (А) показан пример сложной задачи, с которой сталкиваются традиционные модели данных ресурса, который используется в некоторых ссылках предшествующего уровня техники, и при этом две скважины (201-А и 202-А) соединены посредством одного трубопровода с оборудованием (301-А). Датчик давления установлен на считывание и сбор данных на указанном трубопроводе. Как показано в этой системе, две скважины соединены с одним и тем же трубопроводом, и давление трубопровода имеет «тег», который управляется любой сторонней системой. Тег давления трубопровода затем необходимо вручную добавить дважды под каждой скважиной в традиционном способе иерархии. Такой общепринятый способ обеспечивает пользователей неполной и неточной структурой цифрового двойника, поскольку если, например, возникает проблема со скважиной 202-А, которая влияет на данные трубопровода, и пользователь визуализирует данные для скважины 201-А, просматривающий не будет иметь полного понимания того, что происходит с трубопроводом, и не сможет сделать заключение касательно потенциальной первопричины, поскольку могут присутствовать многие первопричины.

[0060] С другой стороны, настоящее изобретение, как показано на фиг. 02 (В), предлагает возможности для создания сложной сети соединений и взаимосвязей, которые обеспечивают целостный и реалистичский общий обзор взаимосвязей, которые

физически существуют между многими наборами данных. Пример считается простым и прямолинейным, но он становится намного более сложным, когда доходит до создания динамических иерархий и соответствующих тегов, которые становятся основанием для других функциональных возможностей или приложений платформы, которые могут включать, например, генерирование динамических диаграмм предельных значений, чтобы показывать ограничения интегрированной производственной системы. В нефтяной и газовой отрасли промышленности производство всегда ограничено наиболее важным узким местом или ограничением (т. е. водой, нефтью или газом). Следовательно, динамическое отслеживание и понимание этих узких мест становится критическим конкурентным преимуществом, подлежащим эффективному использованию. Управляемые вручную и статические иерархии являются подходящими для небольшого промышленного масштаба с ограниченными количествами производственных единиц. Однако при крупномасштабных операциях, в которых существует непрерывный поток миллионов точек данных, подлежащих агрегированию, использование статических иерархий не будет подходить для сложных функциональных возможностей, таких как генерирование таких диаграмм предельных значений.

[0061] В многопрофильной организации те же физические ресурсы отслеживаются в отношении разных контекстов, целей и сфер. Для обслуживания этих различных групп существуют различные иерархии, требуемые для обеспечения правильного анализа для данной сферы или цели. Это может иметь последствия в отношении интеграции данных для сохранения ресурса только раз в базе данных и обеспечения единой версии истины, выполненной с возможностью управления для разных целевых аудиторий. Для обеспечения этого единичные экземпляры корневых записей могут быть связаны с различными иерархиями одновременно. Используя нефтяную и газовую отрасли промышленности в качестве примера, одна скважина может быть рассмотрена посредством разных иерархий. Для операционных ролей пользователи будут рассматривать их скважины в отношении района, тогда как промысловые инженеры хотят рассматривать скважину в отношении резервуара/формации, и инженер по связи будет требовать рассматривать ту же скважину согласно вышке беспроводной связи, через которую к ней проходит связь.

[0062] Иерархии используются для управления доступом к данным к виртуализированным данным из разрозненных организационных источников данных. Абсолютное большинство баз данных, которые используются для таких целей, являются «структурированными», что означает, что они имеют ссылки на столбцы и строки, чтобы сослаться на них в логической таблице. В таких структурах один «атрибут», принадлежащий ко множеству иерархий, необходимо воспроизводить в каждой иерархии (столбце), к которой он принадлежит, и соответственно модифицировать в каждой «иерархии» при внесении изменений. Альтернативно, существуют неструктурированные базы данных (обычно это имеет место в «озерах данных»), в которых по существу хранятся данные с «тегами», в отличие от структурированного способа. Примером этого неструктурированного подхода будут «графовые таблицы». Настоящее изобретение может извлекать (виртуализировать) данные из структурированных таблиц и систем и обрабатывать выходящую информацию, как если бы она поступала из графовой таблицы, чтобы обеспечить возможность простого манипулирования данными в отношении разных требований в платформе. Это позволяет заполнять множество записей из множества разрозненных структурированных таблиц во множество динамических иерархий для конечных пользователей платформы. Следовательно, вместо воспроизведения структурированных иерархий настоящее изобретение изменяет ссылки, чтобы они могли быть обработаны как ссылки с тегами.

[0063] Наиболее часто один и тот же элемент может иметь множество записей, имеющих ту же основную функцию, но содержащих разные значения в одной или множестве баз данных. Посредством сбора данных, когда комбинируется множество запросов считывания/записи, настоящее изобретение добавляет способ в виде функции поиска, который может агрегировать различные  $n$  записей в разных базах данных, несмотря на их разные соглашения о данных, и обеспечивает функциональную возможность для пользователя конфигурировать выбор или исправление «золотой записи» (записи, которая считается наиболее точной для контекста). В качестве типового примера, если вводить название страны в поисковый запрос, результаты должны генерироваться из источника данных, тогда как уникальные идентификаторы, которые показывают перечень стран, должны происходить из другого источника данных. Уникальные идентификаторы элементов в системах распределенных баз данных вызывают проблему целостности данных, которая решается посредством этой золотой записи с повышенным уровнем,

которая проверяется и управляется пользователями. Затем, при обратной записи, данные могут быть отправлены с той же правильной ссылкой на источник, что обеспечивает «совместную работу» разных систем или целостность между ними. Конечные пользователи теперь имеют гибкость при комбинировании правильного источника с правильными данными при необходимости. Посредством этой концепции правильные данные отображаются в правильной иерархии (и как следствие, шаблонах), это обеспечивает то, что этот подход динамически управляет иерархиями для множества сфер, может работать при управлении из слоя платформы конечного пользователя.

[0064] Настоящее изобретение может извлекать и виртуализировать эти «записи», которые могут также называться «элементами» в этом контексте, и управлять ими так, что одна «запись» может быть показана во множестве иерархий/групп. Эта концепция происходит из «графовой базы данных», при этом граф соотносит элементы данных в хранилище с набором узлов и ребер, причем ребра представляют взаимосвязи между узлами. Взаимосвязи позволяют непосредственно связывать вместе данные в хранилище и, во многих случаях, извлекать в рамках одной операции. Графовые базы данных предусматривают взаимосвязи между данными как приоритет. Запрос взаимосвязей становится быстрым, поскольку они постоянно хранятся в базе данных. В дополнение, взаимосвязи могут быть интуитивно визуализированы, что делает их полезными для «сильно взаимосвязанных данных». Следовательно, настоящее изобретение обрабатывает множество распределенных неграфовых баз данных, как если бы они были графовыми базами данных, посредством слоя абстракции данных, затем поднимает уровень возможности конфигурирования выходной информации до конечных пользователей. Более того, иерархия исходной базы данных может быть «смешана и скомбинирована» с другими иерархиями посредством использования способов, подобных графовым базам данных, но, опять-таки, при этом база данных является графовой базой данных.

[0065] В предпочтительном варианте осуществления настоящего изобретения раскрыт реализуемый компьютером способ сбора, интеграции и представления данных. Способ включает создание иерархии атрибутов исходных тегов; и конфигурирование элементов цифрового двойника, чтобы иметь множество атрибутов исходных тегов; при этом иерархии потребителей данных предусматривают динамические критерии сортировки для

всех видов данных, происходящих из хранилища данных и внешних структурированных баз данных, и указанный слой интеграции данных хранит объединенные атрибуты цифровых двойников в базе данных, обеспечивая осуществление обмена той же версии информации с поисковыми запросами и запросами считывания.

[0066] Настоящее изобретение также характеризуется бескодовым способом, в котором используется среда визуальной разработки для обеспечения возможности создания приложений непрофессиональными пользователями посредством таких способов, как перетащить и отпустить, добавление компонентов приложения для создания законченного приложения. Этот бескодовый способ позволяет пользователям с ограниченными предварительными знаниями о кодировании создавать приложения. Настоящее изобретение предоставляет полный набор инструментов для обеспечения разных бескодовых функций для охвата всего цикла создания приложения, а также управления рабочими процессами (бизнес-процессами), использования отображения, конфигурации логики контроля по принципу исключений и конфигурирования модели данных.

[0067] Обычно в общепринятых бескодовых платформах или платформах с низким содержанием кода данные из множества источников импортируются и копируются в базу данных приложения, чтобы они могли быть впоследствии использованы пользователями для создания новых функциональных возможностей. Однако настоящее изобретение обеспечивает бескодовую среду в дополнение к слою абстракции данных, что помогает повысить уровень слоев абстракции сложных и кодированных данных до конечных пользователей, чтобы они могли конфигурировать функциональные возможности без необходимости дублирования данных в базе данных и эффективно использовать возможности виртуализации. Бескодовые среды обычно работают с их собственной внутренней базой данных и должны импортировать данные до того, как данные могут быть использованы в их приложениях для целей, не связанных с кодом. Однако в настоящем изобретении этот промежуток устраняется по мере возможности виртуализации и соединения слоя абстракции данных с организационными базами данных без необходимости импортировать данные в первую очередь. Интерфейс доступен пользователям, чтобы конфигурировать эти соединения/управлять ими, что приводит в результате к полной сквозной бескодовой среде/платформе.

[0068] В других общепринятых способах управления базами данных, созданных для контекстов профессиональных IT администраторов, пользователи используют формы с жесткой кодировкой, которые обеспечивают их возможностью «манипулировать» базой данных посредством предварительно определенных запросов. Единственным способом, которым можно изменить эти формы, является модификация кода, на котором они выполняются. Жесткая кодировка для такой формы займет у опытного разработчика дни работы для создания стабильного процесса запроса, который является неконфигурируемым. Однако настоящее изобретение обеспечивает удобный для пользователя интерфейс и способы для решения таких задач. Например, как показано на фиг. 08-B, пользователи могут использовать бескодовый интерфейс для выбора из множества предварительно установленных соединений, доступных в строке отображения данных. Может быть указан ряд параметров, который обычно будет жестко закодирован. Когда множество различных элементов связаны с одним и тем же «шаблоном», может быть сконфигурирован один запрос «атрибута» (дочерней части элемента), который будет дублирован для всех элементов на основе названия элемента. Название элемента в настоящем изобретении необязательно должно совпадать с названием элемента в базе данных, где запрос будет действовать. Это также означает, что новое оборудование (элементы) и его «атрибуты», добавляемые к платформе, могут быть автоматически сконфигурированы посредством стандартизированного рабочего процесса конфигурирования платформы, который направляет пользователей на различных этапах, необходимых для завершения добавления. Настоящее изобретение предлагает модель данных предварительной конфигурации с жесткой кодировкой, так что фактическая конфигурация может быть бескодовой.

[0069] В предпочтительном варианте осуществления настоящего изобретения присутствует реализуемый компьютером способ отображения данных, причем способ включает: обеспечение множества предварительно установленных соединений в виде инструментов с низким содержанием кода в строке отображения данных; вставку новых функциональных средств и/или изменений в приложение; проверку происхождения требуемых данных из источника базы данных, который уже сконфигурирован в приложении; использование указанных инструментов с низким содержанием кода для конфигурирования новых функциональных возможностей для данных, которые уже

сконфигурированы в приложении; конфигурирование новых соединений посредством базы данных для данных, которые не сконфигурированы в приложении; добавление новых данных в приложение с использованием инструментов конфигурирования данных и посредством виртуализации и параметров эффективности управления; и использование инструментария с низким содержанием кода, легко доступного в приложении, для создания новой функциональной возможности.

### Формула изобретения

1. Реализуемый компьютером способ сбора, интеграции и представления данных, включающий:

обеспечение хранилища данных, слоя абстракции данных, слоя интеграции данных и графического пользовательского интерфейса, при этом происходит обмен набора информации между указанным хранилищем данных и указанным слоем абстракции данных и между указанным слоем абстракции данных и указанным слоем интеграции данных;

хранение временного ряда и реляционных точек данных, собранных с физических ресурсов, в указанных данных из сторонних хранилищ данных или исходных баз данных;

конфигурирование собранных точек данных в цифровые двойники, причем каждый представляет физический ресурс;

выполнение запросов для запроса одного массива данных посредством слоя абстракции данных, который интегрирован в потребителя данных;

комбинирование запросов в массив запросов для уменьшения частоты запросов интерфейса программирования приложений;

применение указанного массива запросов для выбора асинхронных записей и выполнения их потоковой передачи из указанного слоя абстракции в отношении указанного потребителя данных;

обеспечение представления множества асинхронных событий и потоковой передачи указанных событий в отношении потребителя данных; и

отображение графического представления указанных множественных асинхронных событий с применением указанного графического пользовательского интерфейса.

2. Способ по п. 1, отличающийся тем, что указанный потребитель данных дополнительно включает:

обеспечение набора запросов считывания и записи;

генерирование запроса считывания для сбора данных из организационного хранилища данных для обеспечения анализа и отчетов;

генерирование запроса записи для изменения записей, полученных посредством указанного запроса считывания, при этом такой запрос выполнен с возможностью изменения записей в базе данных исходного приложения;

показ обновленной записи посредством нового запроса считывания;

перенаправление указанного запроса считывания автоматически для инициирования нового запроса считывания в базу данных исходного приложения вместо хранилища данных; и

возврат скорректированной и обновленной записи потребителю данных.

3. Способ по п. 1, отличающийся тем, что дополнительно включает:

запись изменений в данные временного ряда или реляционные записи, которые представляют множество физических ресурсов в базе данных;

получение доступа к указанной базе данных для генерирования новых тегов для замен физических ресурсов;

проверку наличия у новых тегов таких же функции и роли, что и у сохраненных в хранилище данных;

конфигурирование указанных новых тегов посредством нового идентификатора версий атрибута, когда новые теги имеют такие же функцию и роль, что и сохраненные в хранилище данных, и связывание всех данных из разных идентификаторов версий атрибутов для представления полного набора данных; и

конфигурирование указанных новых тегов посредством нового идентификатора атрибута, когда новые теги не имеют такие же функцию и роль, что и сохраненные в хранилище данных, и передачу изменений в точки данных временных рядов для одного идентификатора атрибута.

4. Способ по п. 1, отличающийся тем, что указанные цифровые двойники содержат один функциональный элемент и несколько переменных дочерних элементов, которые

сконфигурированы в рамках шаблонов цифровых двойников, при этом один шаблон создается для всех элементов, которые имеют такую же функцию с необязательными атрибутами, где это применимо, конфигурация атрибута анализа для двигательных рабочих характеристик выполняется по меньшей мере один раз без учета других необязательных атрибутов, и разные результаты на основе необязательных атрибутов возвращаются посредством применения формул анализа.

5. Способ по п. 1, отличающийся тем, что дополнительно включает:

создание иерархии атрибутов исходных тегов; и

конфигурирование элементов цифровых двойников для наличия множества атрибутов исходных тегов; при этом

иерархии потребителей данных предусматривают динамические критерии сортировки для всех видов данных, происходящих из хранилища данных и внешних структурированных баз данных, и

указанный слой интеграции данных хранит объединенные атрибуты цифровых двойников в базе данных, обеспечивая то, что происходит обмен той же версии информации с поисковыми запросами и запросами считывания.

6. Реализуемый компьютером способ отображения данных, причем способ включает:

обеспечение множества предварительно установленных соединений в виде бескодовых инструментов в строке отображения данных;

вставку новых функциональных средств и/или изменений в приложение;

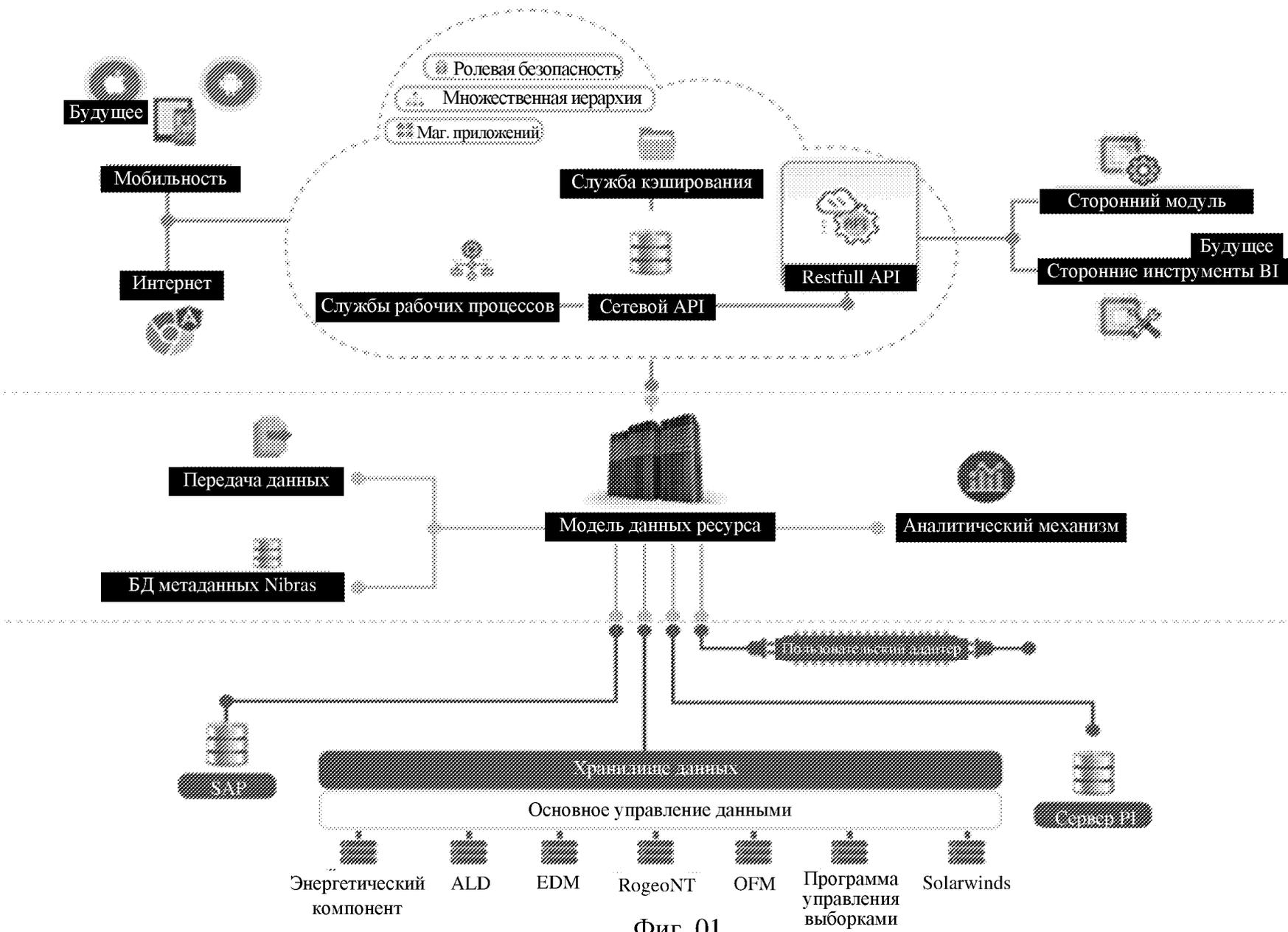
проверку происхождения требуемых данных из источника базы данных, который уже сконфигурирован в приложении;

применение указанных бескодовых инструментов для конфигурирования новых функциональных возможностей для данных, которые уже сконфигурированы в приложении;

конфигурирование новых соединений посредством базы данных для данных, которые не сконфигурированы в приложении;

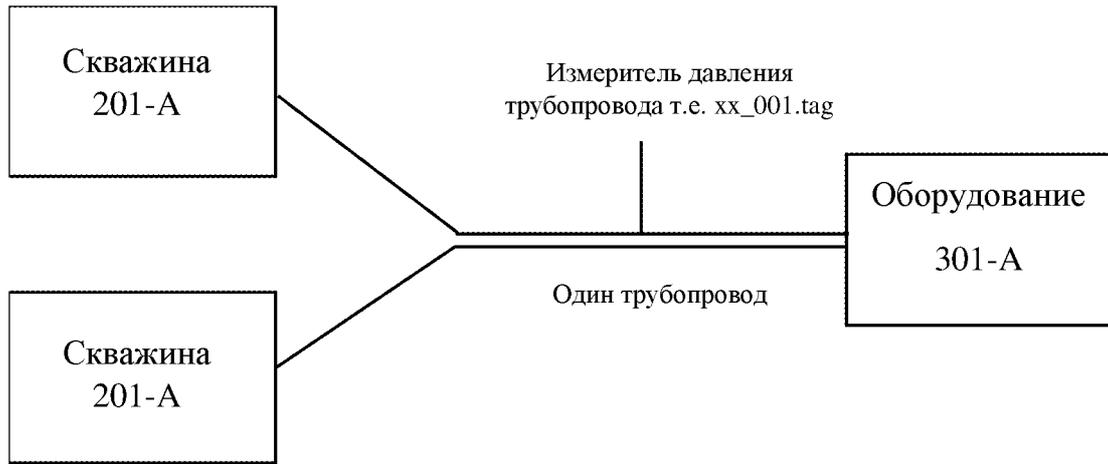
добавление новых данных в приложение с использованием инструментов конфигурирования данных и посредством виртуализации и параметров эффективности управления; и

применение бескодового инструментария, легко доступного в приложении, для создания новой функциональной возможности.



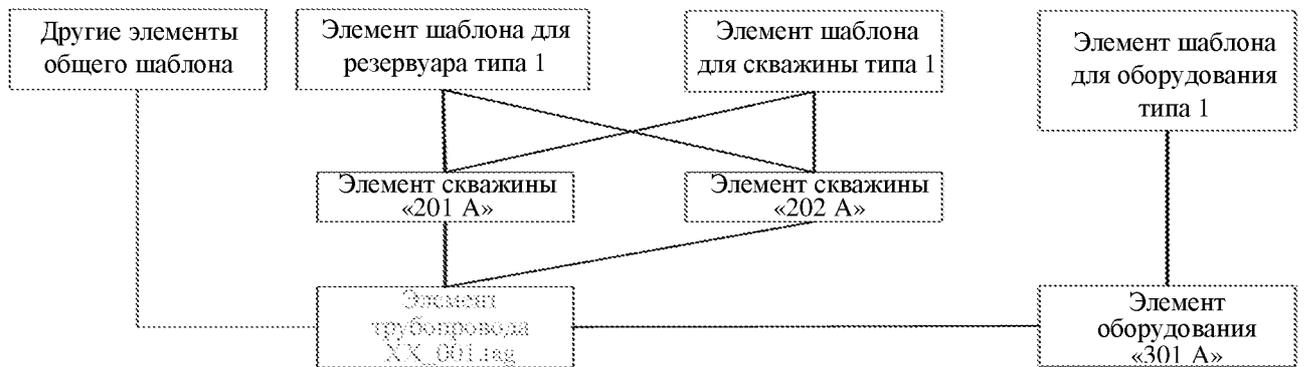
Фиг. 01

(a) Физические взаимосвязи



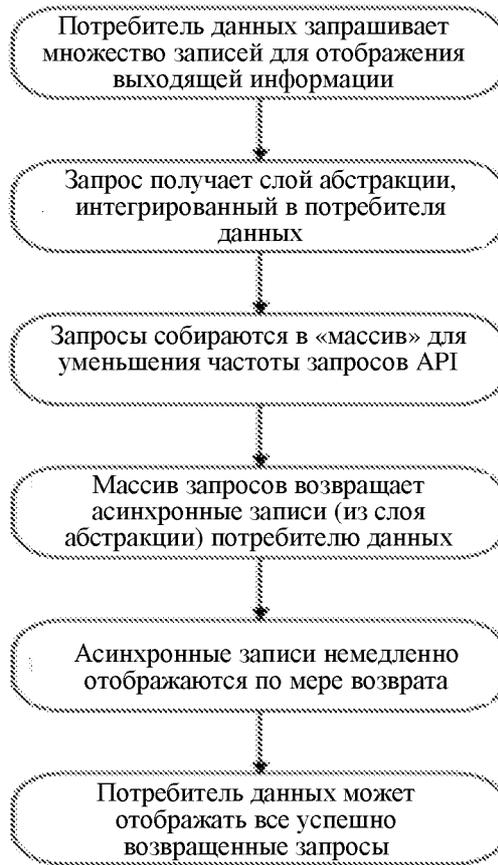
(b) Сложные динамические иерархии

«Динамическая» иерархия Nibras NX

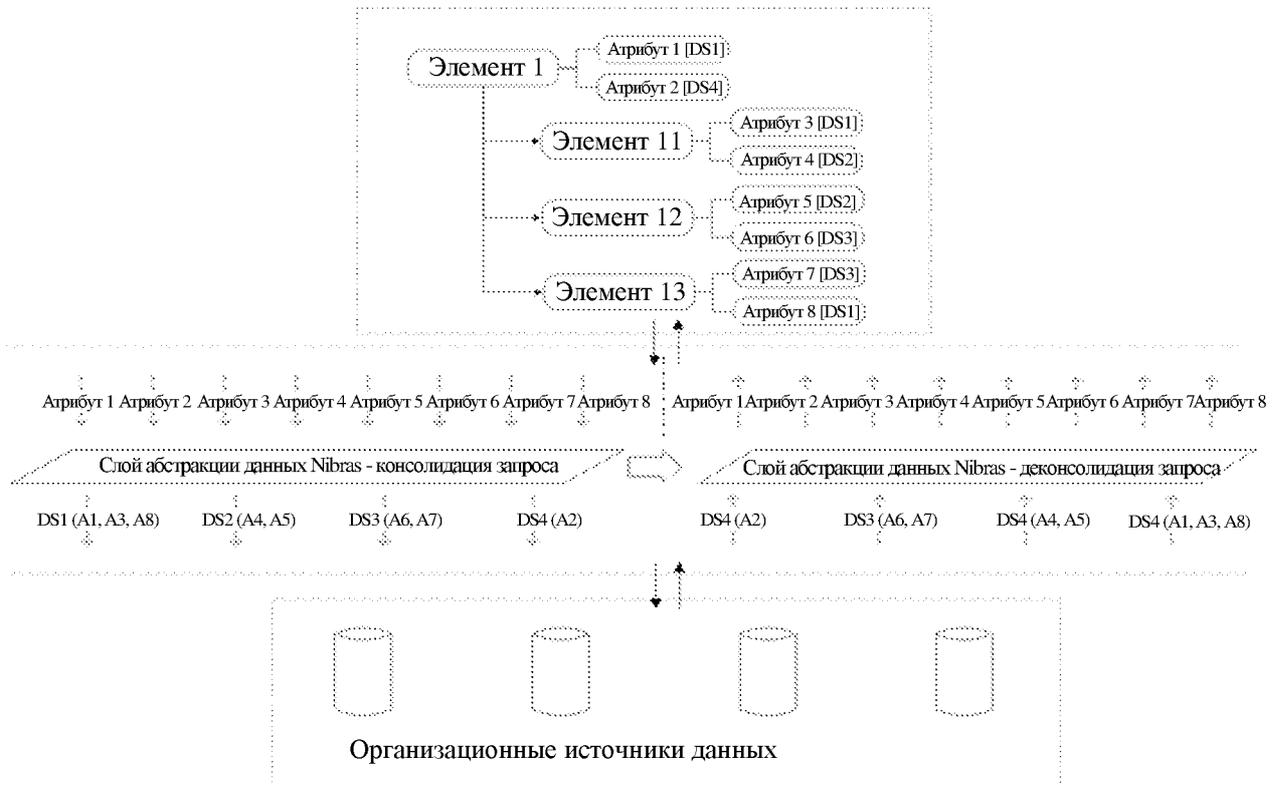


- > Для интегрированной иерархии ресурса
- > Изменение в одном элементе влияет на связанные элементы

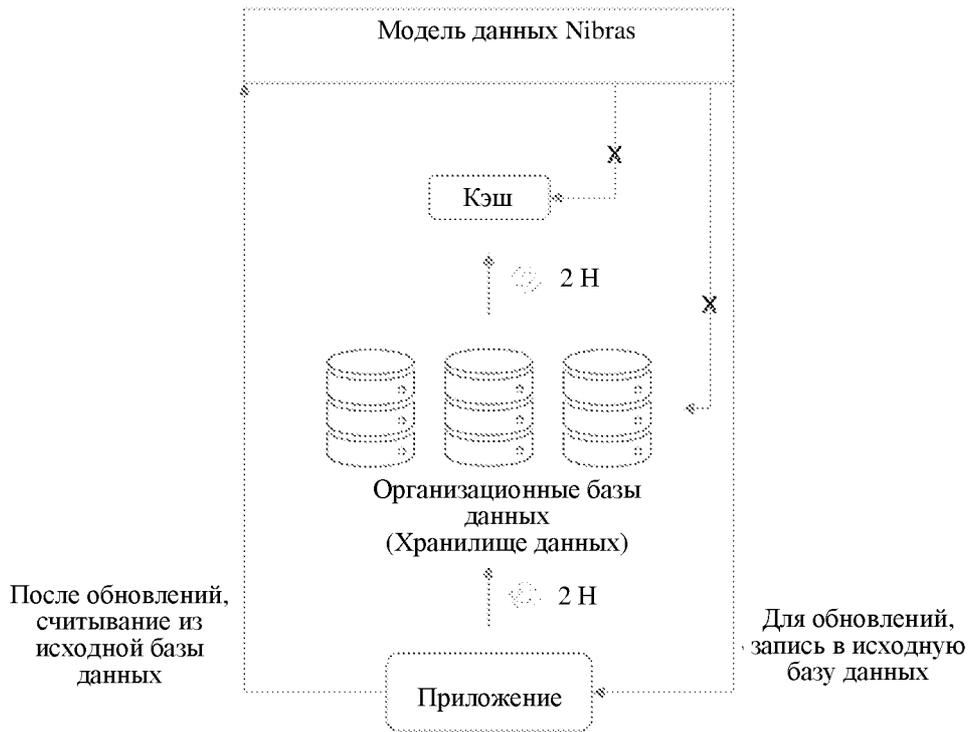
Фиг. 02



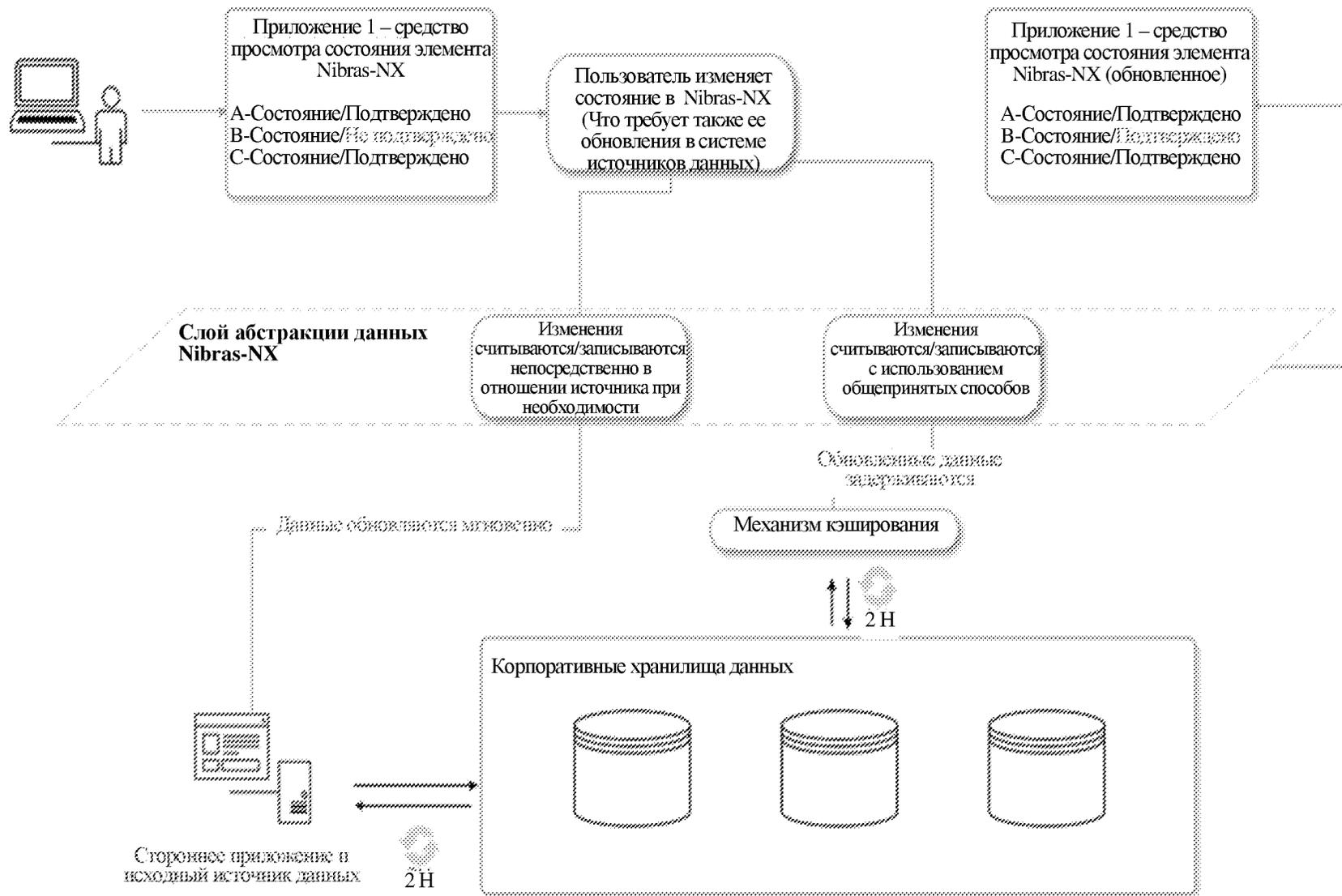
Фиг. 03-А



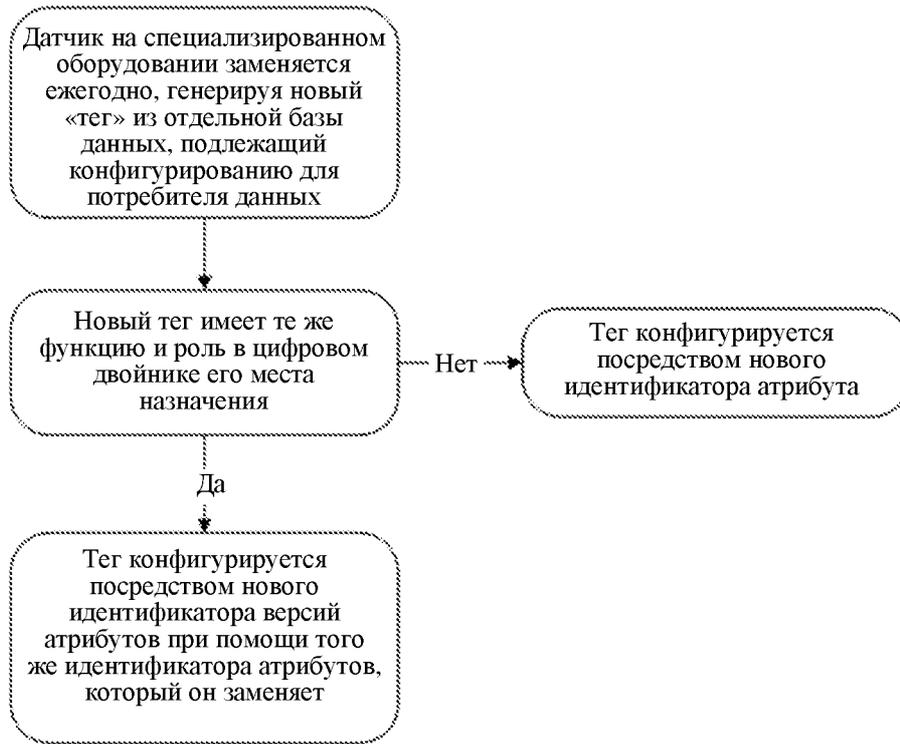
Фиг. 03-В



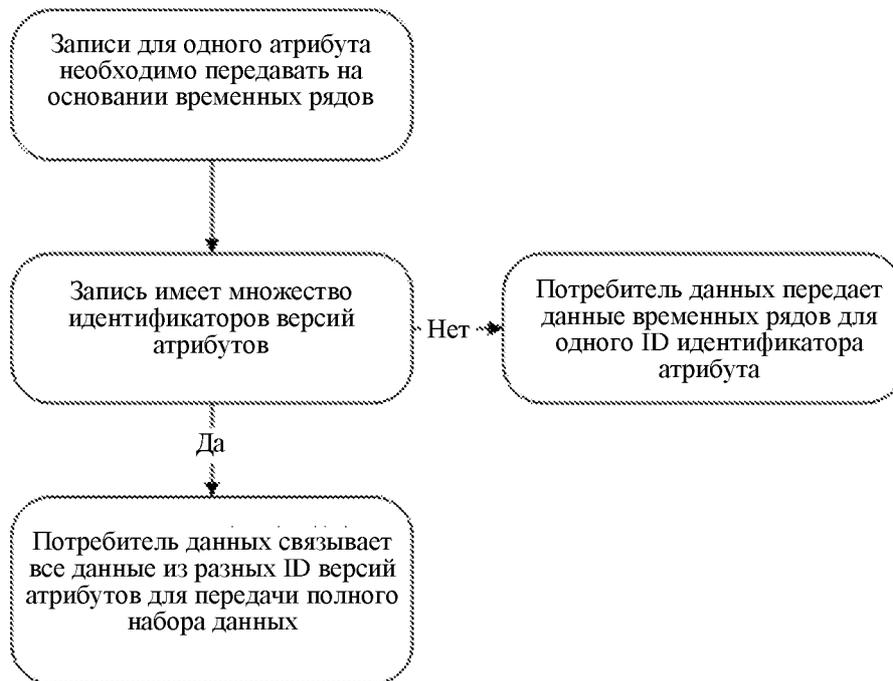
Фиг. 04-А



Фиг. 04-В



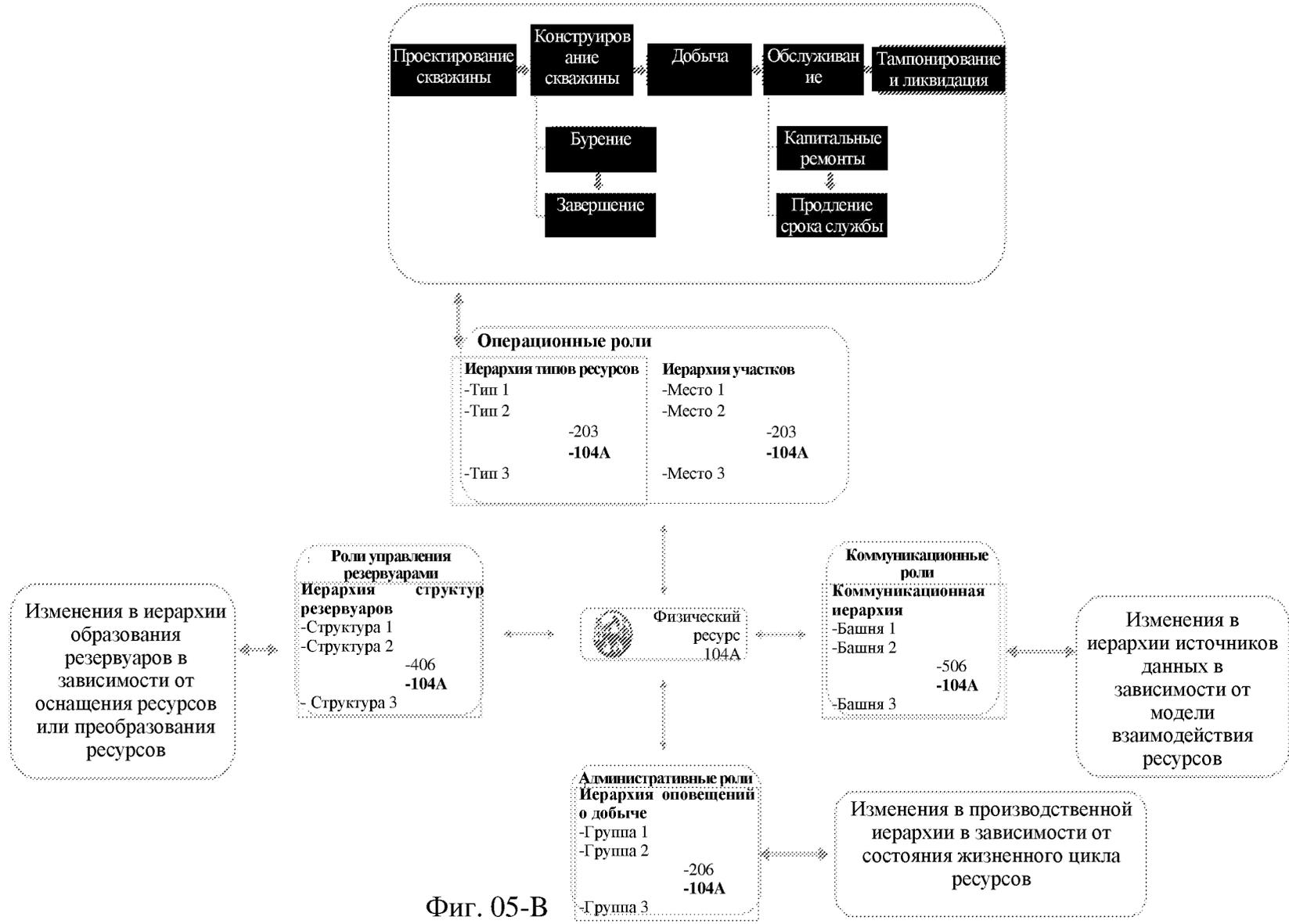
Конфигурирование тега



Комбинирование тегов при анализе и оповещении

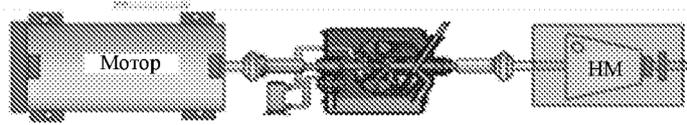
Фиг. 05-А

**Изменения в источниках данных в зависимости от жизненного цикла ресурсов**

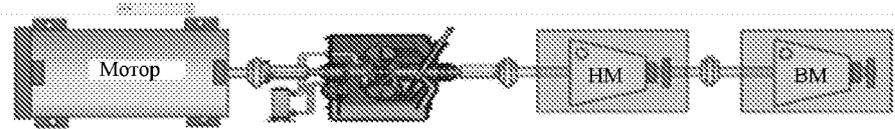


Фиг. 05-В

Компрессор тип 1



Компрессор тип 2



Один функциональный элемент (компрессор) имеет переменные дочерние элементы (тип 1 против типа 2), подлежащие конфигурированию в шаблонах цифровых двойников

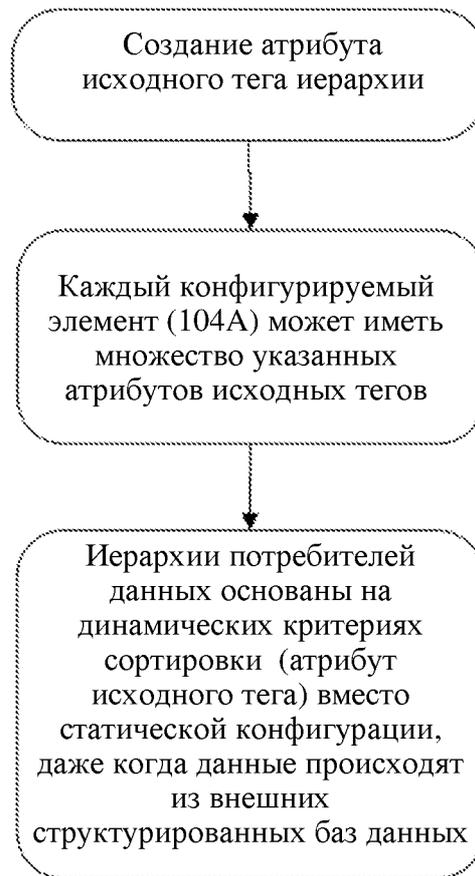
Создание одного шаблона для всех элементов, которые имеют ту же функцию (т. е. компрессор), с «необязательными» атрибутами при необходимости

Требуется конфигурирование атрибута анализа

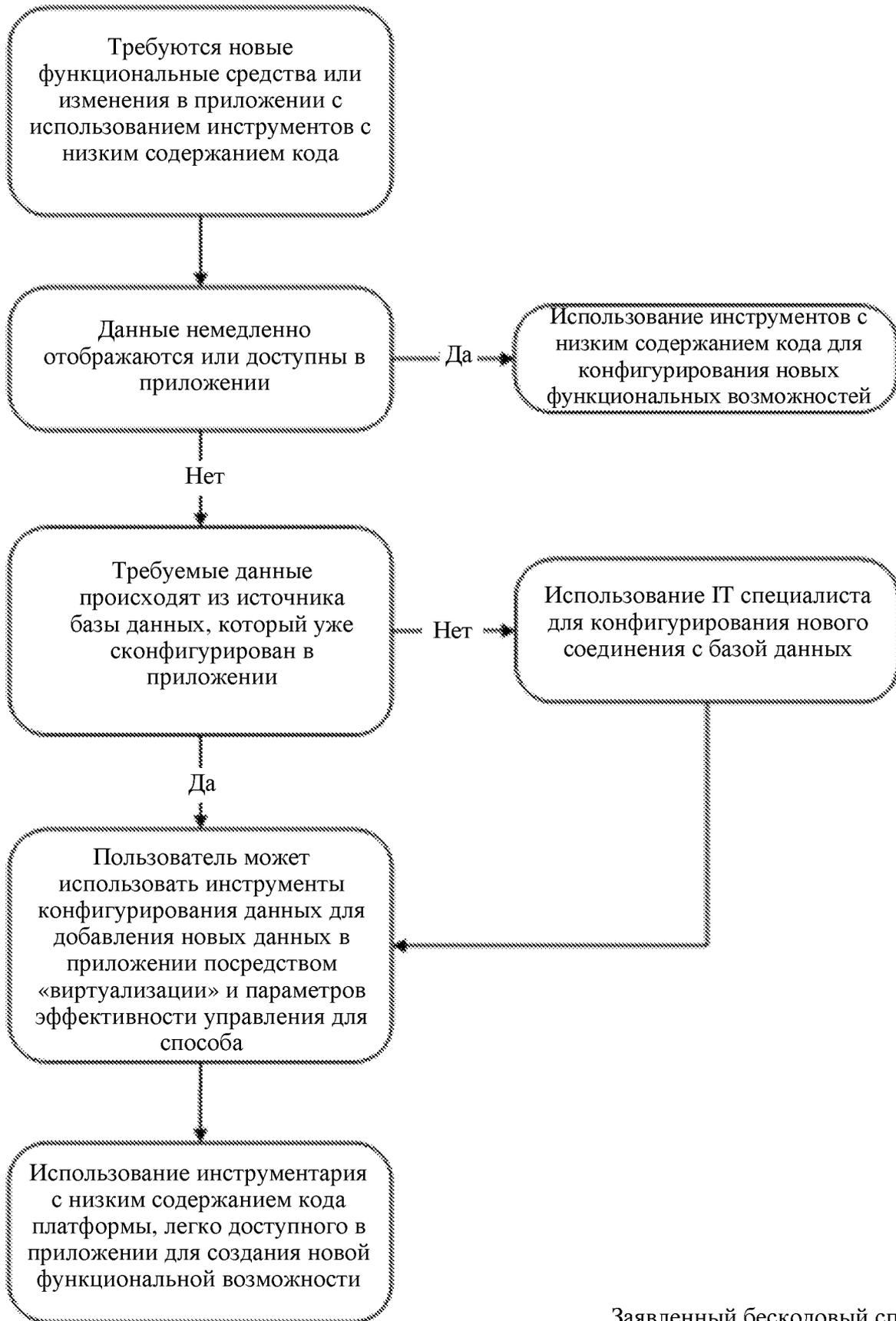
- Конфигурация анализа для рабочих характеристик мотора конфигурируется однажды независимо от необязательных атрибутов
- Формулы для анализа используют булеву логику для возврата разных результатов на основании необязательных атрибутов

Фиг. 06

9/23



Фиг. 07



Заявленный бескодовый способ

Фиг. 08-А

# Template Explorer

STANDARD ELEMENT  
STANDARD

### ATTRIBUTE TEMPLATE DETAILS

NAME *	Commingled Gas Rate %	DESCRIPTION *	Commingled Gas Rate %
LONG NAME		SHORT NAME *	Commingled Gas Rate %
DEFAULT VALUE		VALUE TYPE *	Double
UNITS OF MEASURE	Type UOM or Group of UOM	FRACTIONS DIGITS	0
IS SYSTEM ATTRIBUTE	<input checked="" type="checkbox"/>	ALLOW OVERRIDE	ElementOverride

Tags  +

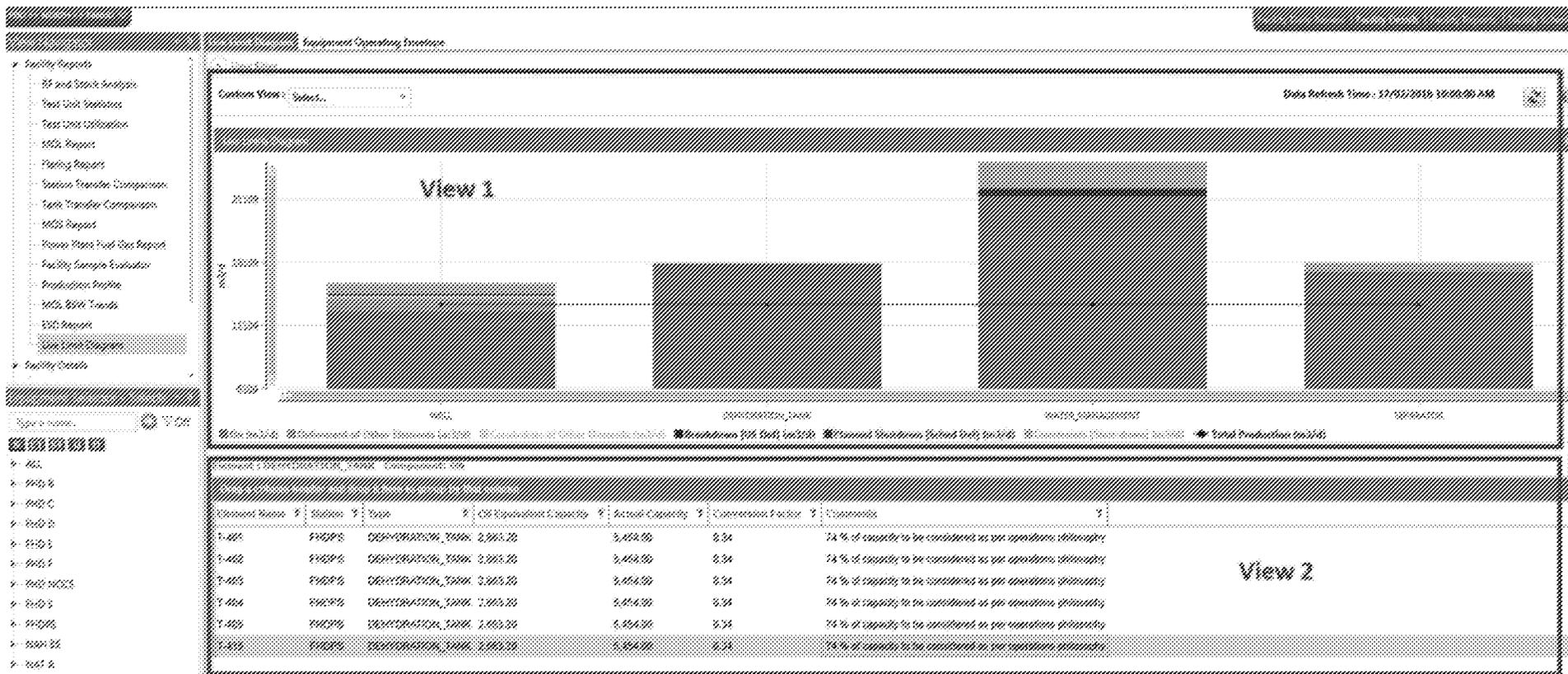
DATA MAPPING

CONNECTION\_QUERY

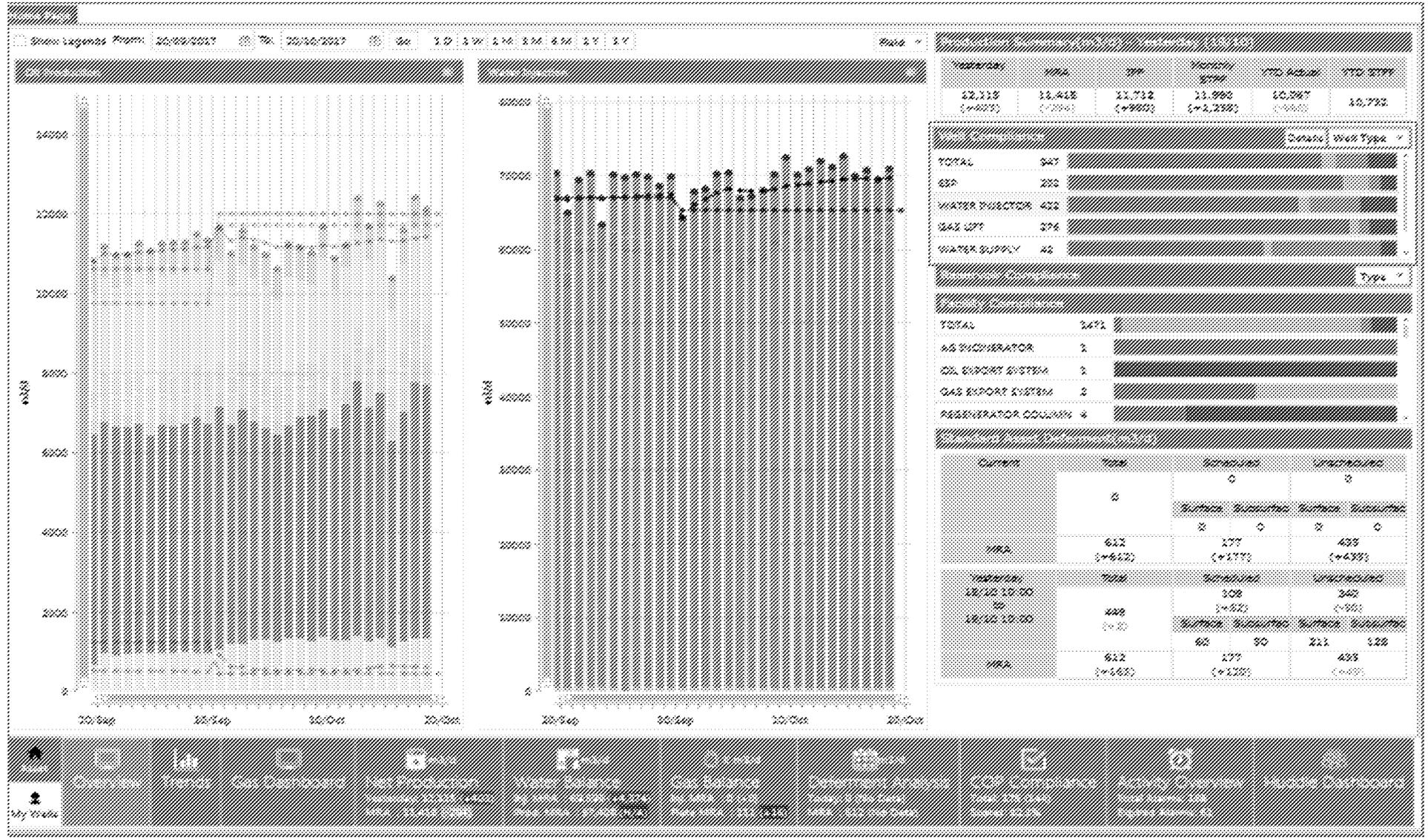
CONNECTIONDR

SourceUOM	Type UOM or Group of UOM
SupportedConfigDataMethods	
ConnectionDR_Inputs (InputParameterAttribute)	
AllowOverride	
AttributePath	
AttributeTemplateVersionID	READ
DefaultElementVersionID	WRITE
ElementFlash	PERSISTENTREAD
ElementTypeID	PERSISTENTWRITE
InputName	WRITEOSOURCE
MetadataLabel	LOOKUP
ParentAttributeTemplateID	
QueryOperation	
QueryOrder	
RootElementTemplateID	

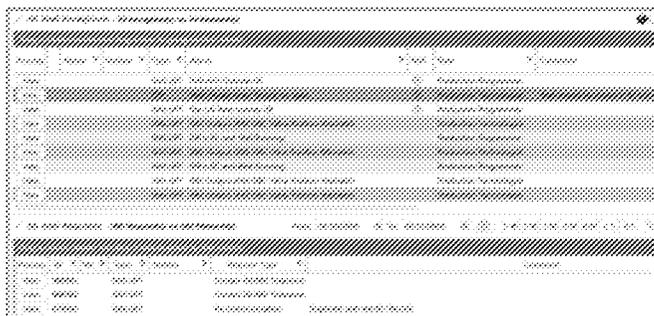
Фиг. 08-B



Фиг. 09



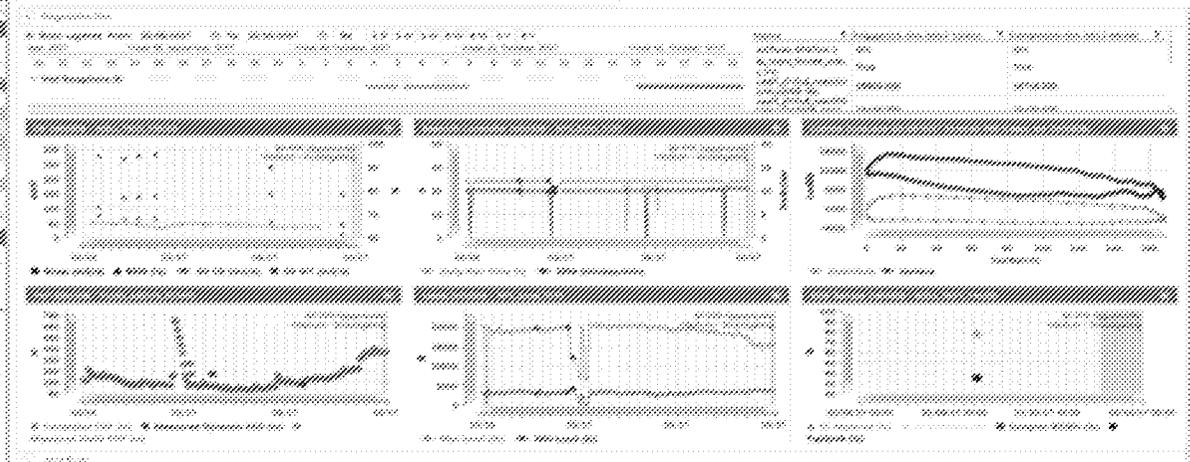
Фиг. 10



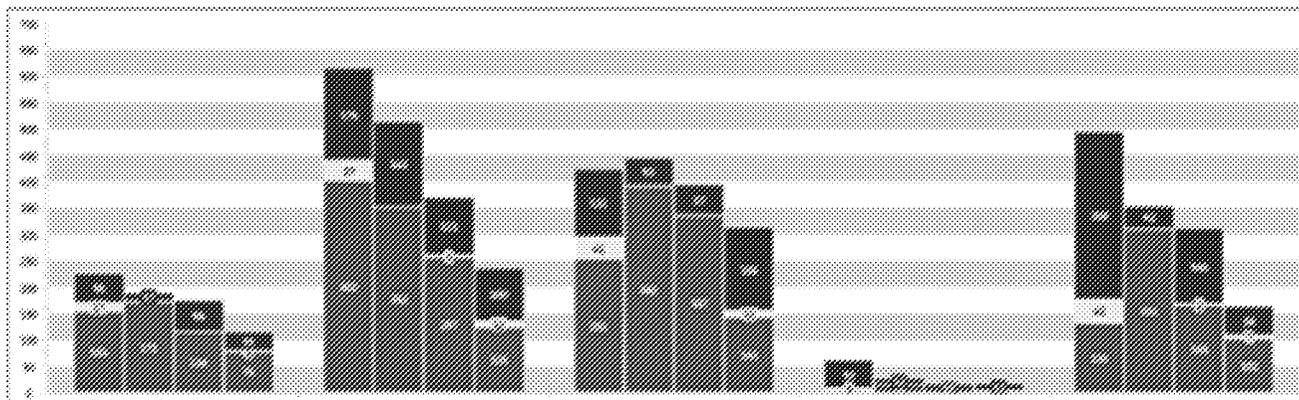
Стандартное средство просмотра исключений определяет приоритетность работы индивидуума на основе значения

Исключение	Приоритет	Статус	Действия
Исключение 1	Высокий	Активен	Действие 1
Исключение 2	Средний	Завершен	Действие 2
Исключение 3	Низкий	В ожидании	Действие 3

SOP направляют инженеров как справляться с каждым исключением в соответствии с минимальным стандартом независимо от уровня опыта.



Стандартные диагностические графики (на каждый тип скважины) для определения первопричины и действия по устранению

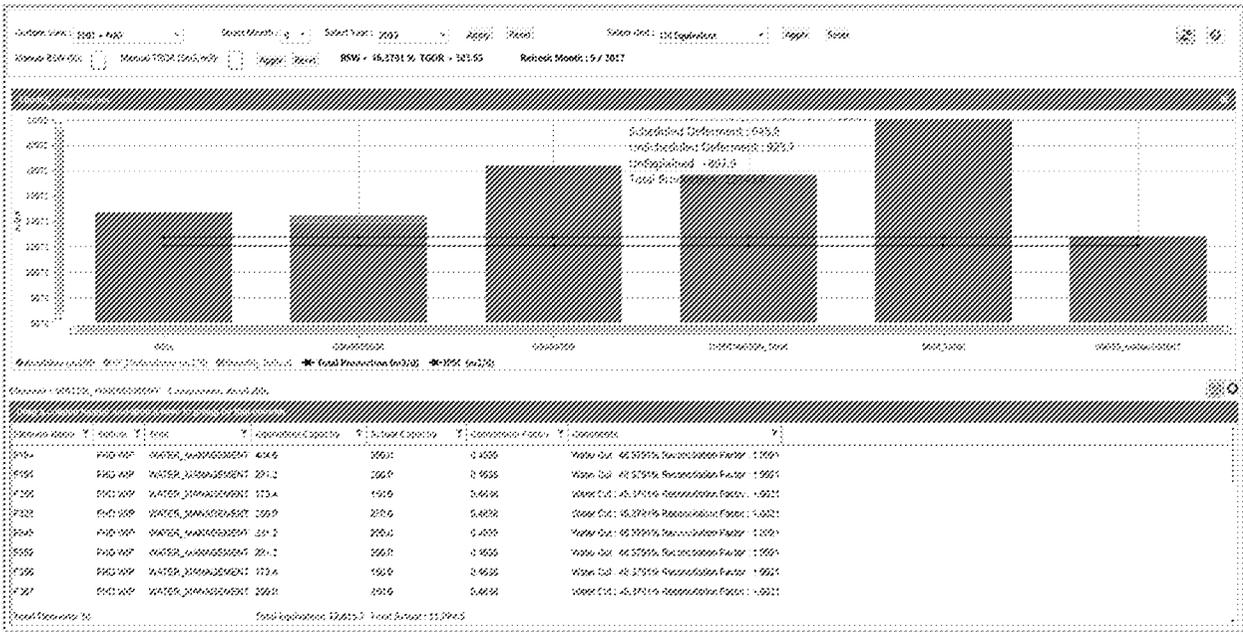


Андон показывает рабочие характеристики обработки исключений в отношении разных ресурсов, типов скважин или ролей

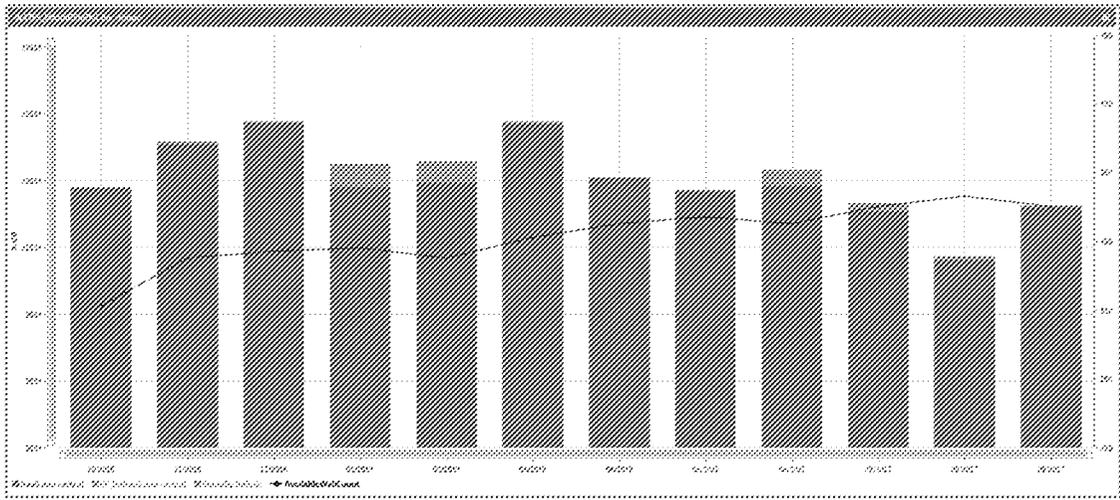
Фиг. 11



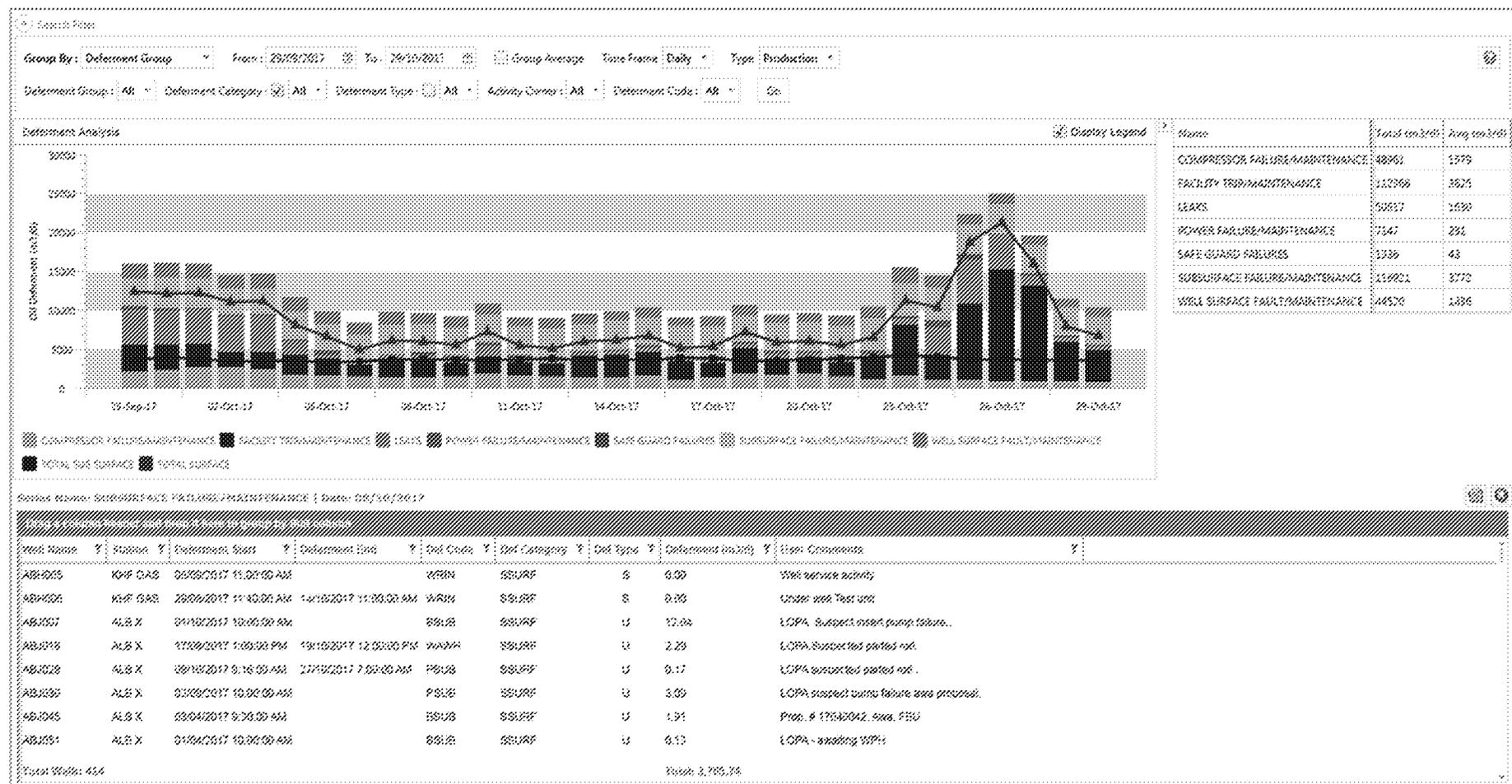
Ежемесячная диаграмма предельных значений в реальном времени



Непрерывная хронология



Фиг. 13



Отчет анализа ежедневных отклонений для добывающих скважин

Фиг. 14

**All Facility Exceptions - 75 Exception(s) on 38 Element(s)**

Process	Name	Station	Type	Alarm	Alarm Owner	Note	Priority	Active (Days)	Expires (Days)	Status	Alarm Type
<a href="#">View</a>	K-0400	FHD-D	CENTRIFUGAL VSD COMPRESSOR	CompressorStatus-Stopped	Com-Mon Reliability North	Reliability Engineer	Medium	36	-32.0	Active	EXCEPTION
<a href="#">View</a>	P-0400	NAT-A	CENTRIFUGAL PSD PUMP	PumpStatus-Stopped	Com-Mon Reliability North	Reliability Engineer	Medium	36	-32.0	Active	EXCEPTION
<a href="#">View</a>	P-04806	FHDPS	CENTRIFUGAL PSD PUMP	PumpStatus-Stopped	Com-Mon Reliability North	Reliability Engineer	Medium	36	-30.9	Active	EXCEPTION
<a href="#">View</a>	K-5705	FHD-F	CENTRIFUGAL VSD COMPRESSOR	Casing_1_Stage_1_Stream_Recycling_With_Low_Surge_Margins	Process Engineer F&HED	Process Engineer	Medium	28	-26.6	Acknowledged Expired	EXCEPTION
<a href="#">View</a>	K-5705	FHD-F	CENTRIFUGAL VSD COMPRESSOR	Casing_2_Stage_1_Stream_Recycling_With_Low_Surge_Margins	Process Engineer F&HED	Process Engineer	Medium	28	-24.9	Active	EXCEPTION
<a href="#">View</a>	K-5705	FHD-F	CENTRIFUGAL VSD COMPRESSOR	Casing_1_Stage_2_Stream_Recycling_With_Low_Surge_Margins	Process Engineer F&HED	Process Engineer	Medium	28	-24.9	Active	EXCEPTION
<a href="#">View</a>	P-04800	FHDPS	CENTRIFUGAL PSD PUMP	PumpStatus-Stopped	Com-Mon Reliability North	Reliability Engineer	Medium	28	-24.7	Active	EXCEPTION
<a href="#">View</a>	K-5701	FHD-F	CENTRIFUGAL VSD COMPRESSOR	CompressorStatus-Stopped	Com-Mon Reliability North	Reliability Engineer	Medium	26	-20.8	Active	EXCEPTION
<a href="#">View</a>	K-5709	FHD-F	CENTRIFUGAL VSD COMPRESSOR	Casing_2_Mon_Drive_End - Potential Compressor Failure - Vibrations Exceeding Max Limits	Process Engineer F&HED	Process Engineer	Medium	26	-23.8	Acknowledged Expired	EXCEPTION
<a href="#">View</a>	K-5709	FHD-F	CENTRIFUGAL VSD COMPRESSOR	Casing_1_Drive_End - Potential Compressor Failure - Vibrations Exceeding Max Limits	Process Engineer F&HED	Process Engineer	Medium	26	-1.4	Acknowledged Expired	EXCEPTION
<a href="#">View</a>	K-5709	FHD-F	CENTRIFUGAL VSD COMPRESSOR	Casing_1_Mon_Drive_End - Potential Compressor Failure - Vibrations Exceeding Max Limits	Process Engineer F&HED	Process Engineer	Medium	26	-5.3	Acknowledged Expired	EXCEPTION

**All Facility Requests - 49 Request(s) on 30 Element(s)**

From: 10/10/2017 To: 29/10/2017

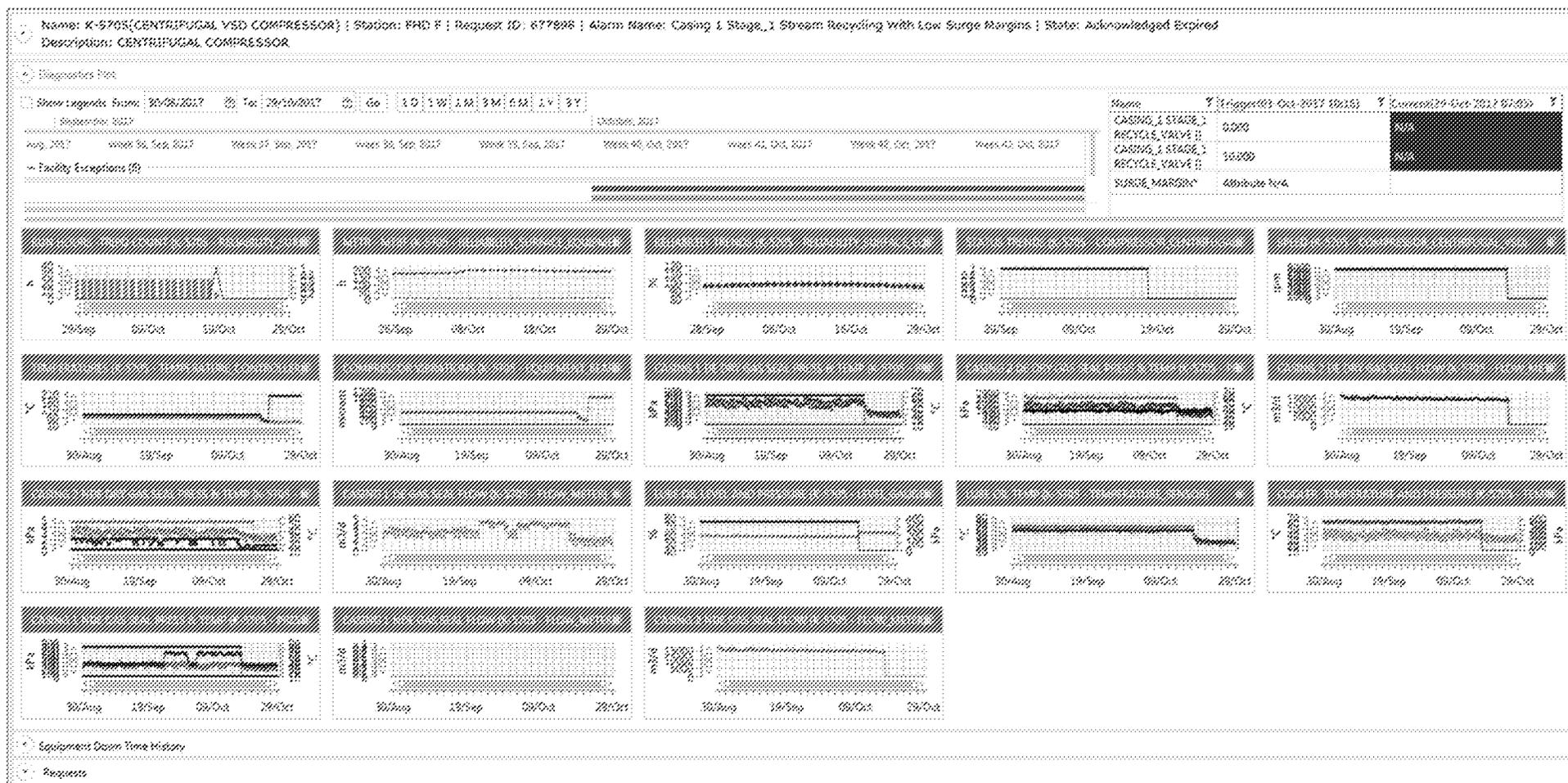
1 W 2 W 1 M 3 M 6 M 1 Y 3 Y

Select Request...

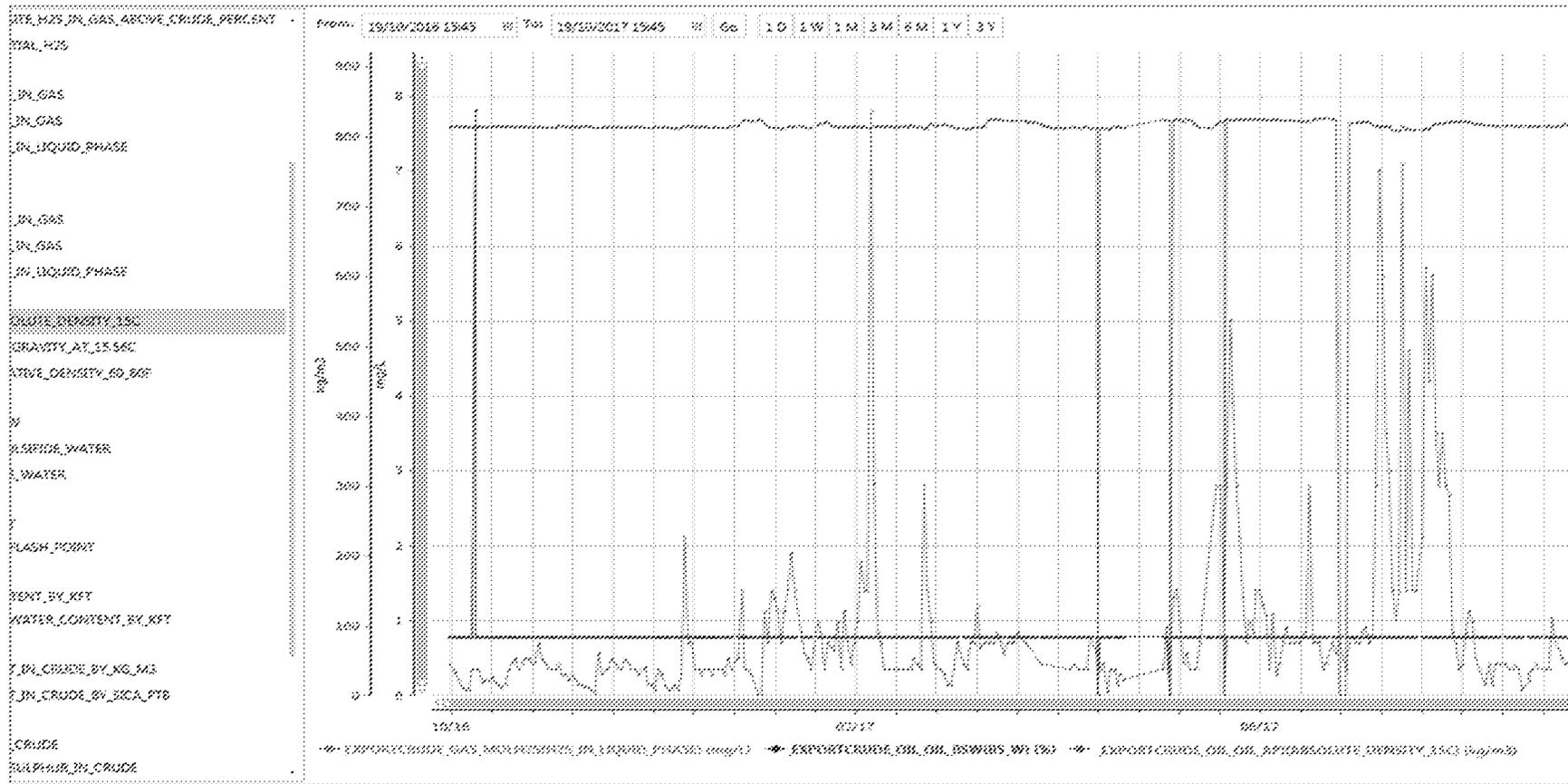
Process	ID	Name	Type	Station	Request Type	Comment	Started On	Updated On	Note	Summary	Expiration	Triggered By	Action
<a href="#">View</a>	690839	V-5713	TWO PHASE VERTICAL SEPARATOR	FHD-C	Update Equipment Configuration		29-Oct-2017 02:57:15	26-Oct-2017 03:09:11	Completed		0	PreConfig	Equipment Configure
<a href="#">View</a>	690468	V-5715	TWO PHASE VERTICAL SEPARATOR	FHD-F	Update Equipment Configuration		25-Oct-2017 02:54:11	26-Oct-2017 02:56:06	Completed		0	PreConfig	Equipment Configure
<a href="#">View</a>	690838	V-5716	TWO PHASE VERTICAL SEPARATOR	FHD-F	Update Equipment Configuration		26-Oct-2017 02:50:36	26-Oct-2017 02:52:17	Completed		0	PreConfig	Equipment Configure
<a href="#">View</a>	690833	V-5708	TWO PHASE VERTICAL SEPARATOR	FHD-F	Update Equipment Configuration		26-Oct-2017 02:44:34	26-Oct-2017 02:46:28	Completed		0	PreConfig	Equipment Configure
<a href="#">View</a>	690832	V-5709	TWO PHASE VERTICAL SEPARATOR	FHD-F	Update Equipment Configuration		26-Oct-2017 02:40:25	26-Oct-2017 02:42:35	Completed		0	PreConfig	Equipment Configure
<a href="#">View</a>	690517	V-5723	TWO PHASE VERTICAL SEPARATOR	FHD-B	Update Equipment Configuration		25-Oct-2017 02:32:51	25-Oct-2017 02:36:25	Completed		0	PreConfig	Equipment Configure
<a href="#">View</a>	690476	V-5724	TWO PHASE VERTICAL SEPARATOR	FHD-F	Update Equipment Configuration		25-Oct-2017 02:08:30	25-Oct-2017 02:23:52	Completed		0	PreConfig	Equipment Configure
<a href="#">View</a>	690479	V-5713	TWO PHASE VERTICAL SEPARATOR	FHD-F	Update Equipment Configuration		25-Oct-2017 02:05:45	25-Oct-2017 02:09:03	Completed		0	PreConfig	Equipment Configure
<a href="#">View</a>	690468	V-5712	TWO PHASE VERTICAL SEPARATOR	FHD-F	Update Equipment Configuration		25-Oct-2017 02:02:13	25-Oct-2017 02:05:52	Completed		0	PreConfig	Equipment Configure
<a href="#">View</a>	690466	V-5711	TWO PHASE VERTICAL SEPARATOR	FHD-F	Update Equipment Configuration		25-Oct-2017 02:01:03	25-Oct-2017 02:04:57	Completed		0	PreConfig	Equipment Configure
<a href="#">View</a>	690465	V-5710	TWO PHASE VERTICAL SEPARATOR	FHD-F	Update Equipment Configuration		25-Oct-2017 02:00:58	25-Oct-2017 02:04:30	Completed		0	PreConfig	Equipment Configure

ES для компрессоров

Фиг. 15-А



Фиг. 15-В



Фиг. 16

All Communication Requests - 517 Requested on 388 Elements

Process	ID	Request Type	Comment	Status	Action
View	645225	Update Well Configuration	well type change from PF to ESP	Completed	Pushed Well In Fieldware
View	645264	Update Well Configuration		Completed	Pushed Well In Fieldware
View	645216	Update Well Configuration	RTU replaced with MC 2	Completed	Pushed Well In Fieldware
View	636328	Update Well Configuration	Please contact Nilras Support to review the existing custom tags	Completed	Pushed Well In Fieldware
View	644680	Update Well Configuration	Please configure THP with this well. Thanks	Completed	Pushed Well In Fieldware
View	654258	Update Well Configuration	we removed the downhole here.	Completed	Pushed Well In Fieldware
View	654269	Update Well Configuration	here we update the flow meter and PIP DCS tag.	Completed	Pushed Well In Fieldware
View	654263	Update Well Configuration	here we update the flowmeter and PIP DCS tag.	Completed	Pushed Well In Fieldware
View	686024	Update Well Configuration	DHG model need to update.	Completed	Pushed Well In Fieldware
View	645082	Update Well Configuration		Completed	Pushed Well In Fieldware
View	645178	Update Well Configuration		Completed	Pushed Well In Fieldware
View	644678	Update Well Configuration	Please configure THP and PIP with this well. Thanks.	Completed	Pushed Well In Fieldware
View	645012	Update Well Configuration		InProgress	Well Test Details
View	643319	Update Well Configuration	Please add the DHG tags.	Completed	Pushed Well In Fieldware
View	643102	Update Well Configuration	Please update configuration.	Completed	Pushed Well In Fieldware
View	625215	Update Well Configuration	back	Completed	Pushed Well In Fieldware
View	643318	Update Well Configuration	Please update the DHG type.	Completed	Pushed Well In Fieldware
View	644055	Update Well Configuration	Please update configuration. OPC mapping not found for MC_RCP_488_ACS_880 connected to RTU_VMBUSIKC-XP in port SERIAL_PORT_DEVICES	Completed	Pushed Well In Fieldware
View	671964	Update Well Configuration		InProgress	Process Request
View	643140	Update Well Configuration	Please update configuration, DHG to well lift and motor controller to ACS PSD	Completed	Pushed Well In Fieldware
View	643284	Update Well Configuration	DHG to be added	Completed	Pushed Well In Fieldware
View	642590	Update Well Configuration	well connect to MSV A-8432 PKT	Completed	Pushed Well In Fieldware
View	642263	Update Well Configuration	Please update configuration, THP to AE-1	Completed	Pushed Well In Fieldware
View	639518	Update Well Configuration	Please update DHG Type	Completed	Pushed Well In Fieldware
View	642993	Update Well Configuration		Completed	Pushed Well In Fieldware

Перечень рабочих процессов конфигурации

Name: (ESP-ON) | Station: | Request ID: 645615 | Request Type: Update Well Configuration

Triggered By: PncConfig

Created By: Started On: 03/06/2023 3:19:00 PM

Executor Role: Executor Group/User:

Due Date: 03/06/2023 10 Days Resource: SEAD

---

**Configuration Details**

Status: Completed

Type: ESP

Area: Pushed Well In Fieldware

Station: Pushed Well In Fieldware

Field: Pushed Well In Fieldware

---

**Communication Details**

RTU Name	RTU	IP Address	Service Port	Channel Name	IOC Switch
MEG226-2	RTU_VMBUSIKC-2	192.168.184.1	502	MEG226_01VMBE_MIS	

---

**Equipment Details**

THP	TUBING_HEAD_PRESSURE	RTU_VMBUSIKC-2_AI_PORT_1	No Scaling
CMP	CASING_HEAD_PRESSURE	RTU_VMBUSIKC-2_AI_PORT_0	No Scaling
DHG	DHG_ESP_WELL_LIFT	RTU_VMBUSIKC-2_SERIAL_PORT_DEVICES	No Scaling
Motor Controller	MC_ESP_PSD_GCS	RTU_VMBUSIKC-2_SERIAL_PORT_DEVICES	No Scaling

---

**Test Skid Details**

MSV: A-

Port: Port 7

Обзор подробной конфигурации

Фиг. 17

**CONFIGURATION**

COMPONENT TIME LINE 1

**DATA MAPPING**

Name	Type	Value
metaLabel		
ParentAttributeTemplateID		

QueryOperation: None, READ, WRITE, PERSISTENTREAD, PERSISTENTWRITE, WRITETOSOURCE

QueryOrder: READ

RootElementTemplateID: WRITE

UOM: P.L.

Aggregate Data Method: WRITETOSOURCE

DataMethod: [ ]

End Time: [ ]

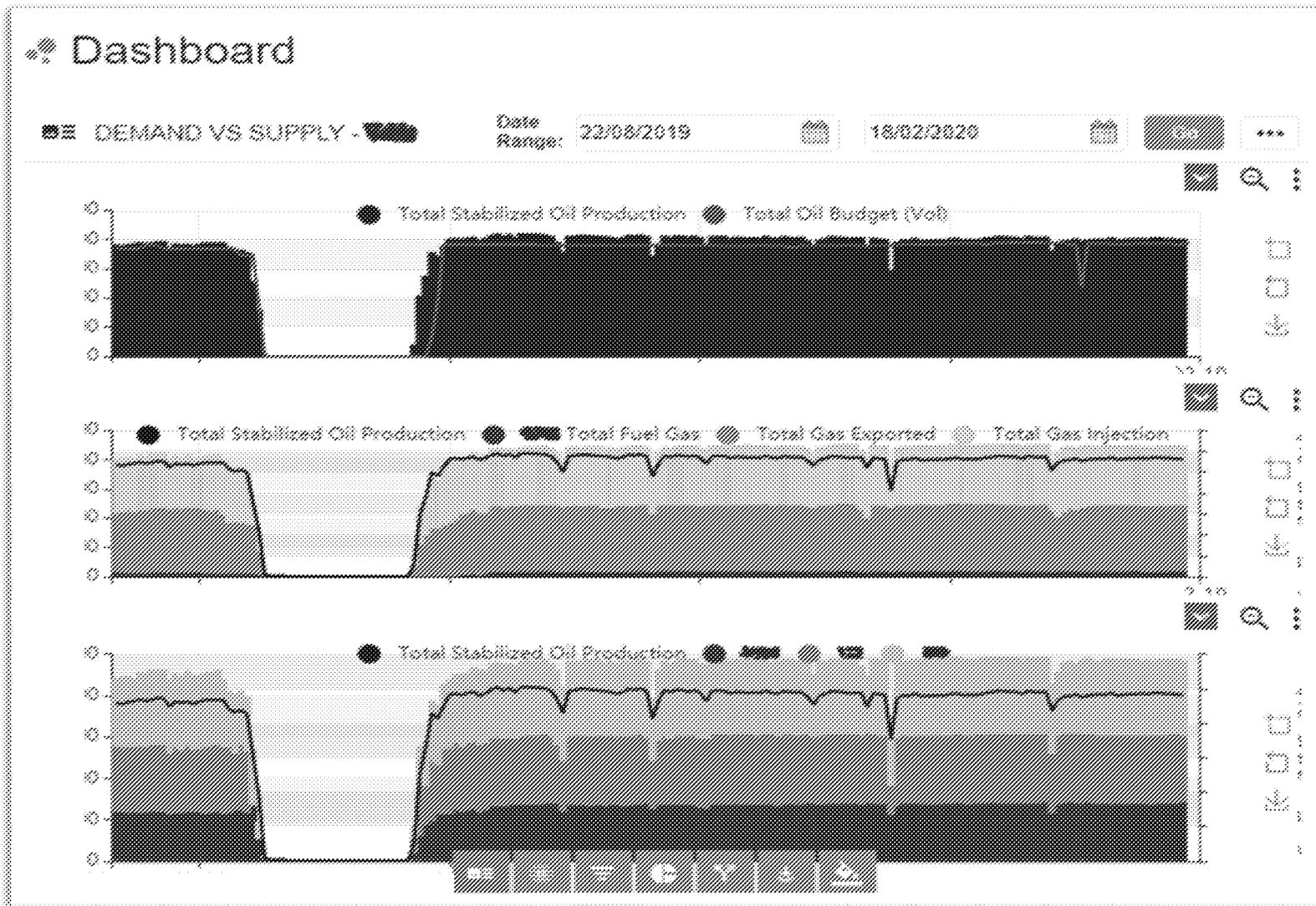
Input name: [ ]

Interval: [ ]

Workflow: Approve, Design Form, Approve Form, Renew Form, Form Published, Comment

Buttons: Save Form, Save Changes

Фиг. 18-А



Фиг. 18-В