

УДК 004.658.6

DOI: 10.15593/2224-9397/2020.3.07

И.А. Шмидт, А.С. ЖелтышевПермский национальный исследовательский политехнический университет,
Пермь, Россия

АВТОМАТИЗАЦИЯ ПОСТЭКСПЕРИМЕНТАЛЬНОЙ ОБРАБОТКИ И ХРАНЕНИЯ ДАННЫХ СТЕНДОВЫХ ИСПЫТАНИЙ

Постэкспериментальная обработка данных, полученных в результате стендовых испытаний, является завершающей и крайне важной стадией испытательного производства сложных изделий. С задачей обработки данных тесно связана задача структурирования и хранения огромного объема данных, порожденных испытаниями. Существующие сейчас способы хранения основаны на ручном структурировании данных испытаний. Большую часть подлежащей накоплению информации составляют результаты наблюдения за параметрами процесса испытания, так называемые тренды. При наблюдении значения параметров фиксируются и привязываются к моменту наблюдения, т.е. являются временными рядами. Использование для хранения таких данных традиционных реляционных систем управления базами данных (РСУБД) является неоправданным, так как они не приспособлены для эффективного хранения и обработки временных рядов. **Цель исследования:** разработка системы сбора, обработки и хранения данных. **Методы:** выбор и обоснование в качестве альтернативы РСУБД для хранения данных испытаний документоориентированной БД. Создание комплекса приложений – экосистемы хранения и обработки данных, предназначенного для импорта данных в систему и их обработки. **Результаты:** на базе предложенного метода хранения данных в документоориентированной БД были разработаны структуры, обеспечивающие хранение всей сложноорганизованной информации об испытаниях. Разработан программный комплекс, реализующий такие функции системы, как: импорт данных испытаний в Базу данных, просмотр тренда, параметров, ассоциированных документов, поиск и извлечение данных. Наиболее востребованной пользователями функцией обработки данных испытаний является поиск характерных особенностей поведения изделия, т.е. анализ трендов. В качестве событий, характеризующих поведение тренда, рассматриваются такие, как попадание значения параметра (или его производной) в заданный диапазон, достижение минимума/максимума значения параметра, переключение состояния дискретного параметра и т.д. Предложен способ организации сложных поисковых запросов, объединяющих описание нескольких событий через И, ИЛИ. **Практическая значимость:** разработанная единая система сбора, обработки и хранения данных о проведенных испытаниях предоставляет специалистам по обработке данных мощный инструмент анализа.

Ключевые слова: испытания сложных технических изделий, документоориентированная СУБД, временные ряды.

I.A. Shmidt, A.S. Zheltyshev

Perm National Research Polytechnic University, Perm, Russian Federation

AUTOMATION OF POST-EXPERIMENTAL PROCESSING AND STORAGE OF BENCH TEST DATA

Post-experimental processing of data obtained as a result of bench tests is the final and very important stage in the test production of complex products. Closely related to the task of data processing is the task of structuring and storing the huge amount of data generated by tests. Current storage methods are based on manual structuring of test data. Most of the information to be accumulated consists of the results of monitoring the parameters of the test process, the so-called trends. When observing, the values of the parameters are fixed and tied to the moment of observation, i.e. are time series. The use of traditional RDBMS for storing such data is unjustified, since they are not suited for efficient storage and processing of time series. **Research objective:** development of a system for collecting, processing and storing data. **Methods:** selection and justification as an alternative to RDBMS for storing test data of a document-oriented database. Creation of a complex of applications – an ecosystem of data storage and processing, designed to import data into the system and process it. **Results:** on the basis of the proposed method of storing data in a document-oriented database, structures were developed that provide storage of all complex information about tests. A software package has been developed that implements such system functions as: importing test data into the Database, viewing a trend, parameters, associated documents, searching and extracting data. The most demanded by users function for processing test data is the search for characteristic features of product behavior, i.e. analysis of trends. As events characterizing the behavior of a trend, such events as falling of a parameter value (or its derivative) into a given range, reaching a minimum/maximum of a parameter value, switching a state of a discrete parameter, etc. are considered. A method is proposed for organizing complex search queries that combine the description of several events using AND, OR. **Practical significance:** the developed unified system for collecting, processing and storing data on the tests performed provides data processing specialists with a powerful analysis tool.

Keywords: testing of complex technical products, document-oriented DBMS, time series.

1. Цель и методика проведения испытаний. Испытание – это экспериментальное определение количественных и (или) качественных характеристик свойств объекта испытаний как результата воздействия на него при его функционировании, при моделировании объекта и (или) воздействий [1]. Испытание является крайне важной стадией технологического цикла продукции, так как позволяет выявить показатели качества конечного продукта и их соответствие предъявленным требованиям. Для проведения испытаний продукции в машиностроительной отрасли создаются целые испытательные производства. Они обеспечивают основные виды испытаний: доводочные, предварительные, приёмо-сдаточные и др.

Вкратце, на сегодняшний день стендовое испытание сложного технического изделия (СТИ) выглядит следующим образом. Изделие помещается на стенд, далее создаются условия, приближенные к экс-

плуатационным или превышающие их (это зависит от вида испытания). В это же время с определённой частотой опроса по каналам измерения поступают данные с нижнего уровня (датчиков) на верхний уровень (контроллер). Эти данные сохраняются в виде бинарных файлов – так называемых трендов. Информация о каналах измерения (обозначение измеряемого параметра, допустимая погрешность, наименование датчика и др.) записывается в отдельный конфигурационный файл – паспорт. Затем, после окончания испытания, тренды и все остальные данные об испытании и его объекте поступают в специальное подразделение, которое занимается обработкой этих данных и исследованием причин дефектов изделия.

Остановимся поподробнее на последнем этапе. Бинарный файл тренда конвертируется в формат профессионального пакета постэкспериментальной обработки измерительной информации WinПОС (MERA) [2]. Таким образом, основной объём данных о поведении испытываемого изделия хранится в специфическом формате MERA.

После обработки тренда делается расчёт параметров. Расчёт представляет собой вычисление показателей качества изделия (например, КПД) на основе измеренных параметров (например, число оборотов в минуту). Результаты расчётов сохраняются в файле протокола, протокол является Excel-документом, который содержит в себе метаинформацию об испытании, измеренные и расчётные параметры на точках (режимах), графики основных расчётных параметров, заключение по параметрам.

2. Оценка объёма хранимых данных. Результаты всех проведённых испытаний являются ценными с точки зрения аналитики и предоставления отчётности предприятия, поэтому их необходимо сохранять. Проведём оценку объёма данных, подлежащих хранению. Большую часть хранимой информации представляют тренды испытаний. При наблюдении значения параметров фиксируются и привязываются к моменту наблюдения, т.е. являются временными рядами [3, 4].

Для примера рассмотрим испытания турбин. В среднем, чтобы испытать одну турбину, делается 2–6 запусков. В каждом запуске регистрируется порядка 130 параметров, при этом в тренд записывается около 50 000 замеров на каждый параметр, таким образом, объём хранимых данных для каждого тренда составляет около 60 Мб. Помимо тренда около 10 Мб отводится на другие файлы: протокол, паспорт и различные конфигурационные и журнальные файлы.

На текущий момент испытано около 80 турбин. Примерно каждый месяц испытывается новая турбина. С расчётом на 10 лет для хранения всех данных потребуется около 100 Гб.

3. Способ хранения данных. Все эти файлы о проведённых испытаниях (тренды, протоколы, паспорта, файлы WinПОС, коэффициенты калибровки, фотографии видимых дефектов образца и т.д.) на сегодняшний день хранятся только в файловой системе. Следовательно, для каждого испытанного изделия информация о проведенных работах размещается в некоторой типовой структуре каталогов (рис. 1).



Рис. 1. Данные испытаний в файловой системе

Структура подкаталогов отражает процесс испытания каждого изделия. СТИ испытывается сериями работ – постановками на стенд. В разных постановках могут использоваться разные составные элементы изделия. В примере с турбиной это статор, ротор и коллектор.

Основной же единицей испытания является работа – запуск изделия. В директории каждой работы хранятся тренд, протокол, коэффициенты калибровки, каталог файлов MERA и другие файлы.

Для описания всего набора проведенных испытаний как единого целого создается файл в формате Excel, так называемый «Контроль стабильности». Он хранит в себе метаинформацию по каждой проведённой работе: дату, номер протокола, номера составных частей испытанного изделия, вид испытания, условия проведения, заключение и др.

После проведения очередной работы на испытательном стенде в существующую систему хранения вносятся новые данные. Делается это следующим образом: создаётся новая директория в папке со всеми испытаниями, туда помещаются все файлы с рабочей станции стенда; в ней же создаются файл протокола испытания и MERA-файлы. Затем вносятся изменения в файл контроля стабильности.

Всё это делается вручную, т.е. имеет место человеческий фактор. Это один из недостатков существующего способа хранения.

4. Проблемы имеющейся технологии хранения и обработки данных. У существующего способа хранения есть ряд недостатков:

– разделение и изоляция данных [5]. Поскольку данные хранятся в разных файлах, то собрать информацию из разных файлов достаточно сложно, приходится создавать некоторый временный файл, в который собирается вся необходимая информация, так как приходится организовывать синхронную обработку данных сразу в нескольких файлах;

– дублирование данных. Из-за того, что данные хранятся в разных файлах, очень часто приходится повторять некоторую информацию, чтобы не потерять смысл;

– зависимость от данных. В файловых системах все данные, а также их структура жестко прописаны, поэтому любое изменение в данных или в структуре требует значительных усилий для внесения такого изменения;

– человеческий фактор. Вследствие него могут быть нарушены установленная структура подкаталогов и правила именования файлов. Это может затруднить автоматизированную обработку;

– отсутствие механизма поиска данных. Проблема, с которой сталкиваются инженеры при обработке архивных данных испытаний, – это поиск информации об этих испытаниях по определённым критериям, которые затрагивают параметры, хранящиеся в файлах трендов.

Например, необходимо вывести номера протоколов и даты всех испытаний, в тренде которых параметр N превышал значение 15 000. Для этого потребуются найти все файлы трендов испытаний и каким-то образом сопоставить их с файлами протоколов. В файловой системе это сделать проблематично, так как тренд и протокол могут вследствие человеческого фактора быть расположены в разных директориях и под разными названиями от испытания к испытанию.

5. Концепция системы сбора, обработки и хранения данных, требования к системе. Для устранения всех вышеперечисленных недостатков предлагается создать единую систему сбора, обработки и хранения данных о проведённых испытаниях, т.е. можно сформировать цель исследования как разработку системы сбора, обработки и хранения данных.

Разрабатываемая система должна удовлетворять определённым требованиям. Основное требование к системе – это решить существующие проблемы хранения послеэкспериментальных данных испытаний. Для этого необходимо хранить данные в единой БД, под которой будем понимать совокупность программ и языковых средств, предназначенных для управления данными и обеспечения взаимодействия их с прикладными программами (ГОСТ 20886-85). Отсюда вытекает следующее требование – сам переход от существующего способа хранения. Все накопленные данные в файловой системе нужно перенести в БД, т.е. в системе должен быть реализован механизм импорта.

Далее, после того как выбран способ хранения, требуется разработать функционал поиска и отображения испытаний по различным параметрам. Это могут быть как параметры самого испытания, так и параметры тренда.

Для проведения обработки послеэкспериментальных данных необходимо реализовать автоматизированное выделение режимов испытания и расчёт параметров на этих режимах. Автоматизированное выделение означает, что все режимы, которые явно представлены в тренде в каком-либо параметре, система должна распознавать. Пользователь, в свою очередь, должен иметь возможность редактировать набор режимов.

Функционал проведения расчётов должен включать в себя хранение и обработку данных расчётных параметров – как минимум, обозначений и расчётных формул. Используя эти данные, система должна автоматически вычислять и отображать значения расчётных параметров на каждом режиме.

Также требуется, чтобы система имела возможность интеграции с другими программами, участвующими в технологическом процессе обработки послеэкспериментальных данных. Для этого нужно разработать алгоритмы экспорта в такие форматы, как MERA (для экспорта тренда) и CSV (для информации, представленной в таблицах).

6. Выбор и обоснование способа хранения, обзор имеющихся подходов. Проблему выбора адекватного задаче способа хранения можно считать самостоятельной научной проблемой.

Осознание важности проблемы хранения данных испытаний сложных изделий и построение такой системы применительно к процессу испытаний газотурбинных двигателей (ГТД) представлено в [6]. Вся информация о стендовых испытаниях ГТД сохраняется в базе данных на основе языка разметки XML. Такой способ хранения хорошо подходит для сохранения слабо структурированной, разнородной информации об изделии и об условиях и конфигурациях испытаний, но неудобен для записи переходных режимов, т.е. собственно трендов.

Есть две проблемы при выборе способа хранения данных об испытаниях. Первая проблема – это то, что временные ряды неудобно хранить в традиционных реляционных СУБД (РСУБД). Существует класс специализированных баз данных временных рядов (Time-series databases) [7, 8]. Большинство таких БД организовано в структуры, которые записывают значения для одного элемента. Например, можно создать таблицу для отслеживания температуры процессора. Внутри каждое значение будет состоять из временной метки и показателя температуры. В таблице может быть несколько метрик. Таким образом, эти базы не приспособлены для хранения какой-либо другой информации, не являющейся временными рядами. А в нашем случае временные ряды являются важной, но не единственной информацией, описывающей испытания.

Второй проблемой является то, что для хранения информации об испытаниях РСУБД приспособлены плохо, так как набор регистрируемых параметров (столбцы в таблице данных) меняется от испытания к испытанию. Также РСУБД плохо подходят для сохранения разнородных дополнительных данных (фото, файлы документов и пр.).

Проблема, заключающаяся в том, что каждая серия экспериментов имеет свой набор разнообразных измеряемых и вычисляемых параметров, что существенно усложняет создание единой структуры БД, была отмечена в [9]. Для решения этой проблемы был предложен вариант организации данных, когда база содержит только однотипную для всех экспериментов информацию, а остальные данные сохраняются в прикрепленных файлах на сервере.

Эта же проблема в [10] решается так: все данные всех параметров записываются в одну огромную таблицу, имеющую следующую структуру полей:

- testing_data_id – уникальный идентификатор;
- testing_id – идентификатор испытания;
- parameter_id – идентификатор параметра;
- Value – данные тестирования.

Учитывая проведенную выше оценку объема хранимых данных, количество записей в этой таблице для нашего случая превысит 10 миллиардов, что делает такую реализацию явно неработоспособной.

Как альтернатива РСУБД предлагается использование документоориентированной СУБД MongoDB [11], которая обладает следующими преимуществами:

- возможно хранение больших объемов неструктурированной информации в отличие от РСУБД;
- не загромождена механизмом контроля целостности данных, который является избыточным для решения нашей задачи;
- предоставляет развитые механизмы запросов;
- большие возможности резервного копирования (репликация);
- балансировка нагрузки (шардинг);
- базовая (community) версия является бесплатной.

Способ хранения временных рядов в документоориентированной базе данных рассмотрен в [12–15].

Практическая реализация выбранного способа хранения, как показано ниже, подтвердила правильность сделанного выбора.

7. Использование JSON-сценария импорта. Первой задачей, возникающей при создании предлагаемой системы, является импорт уже накопленных данных об испытаниях в документоориентированную БД. Эти данные не имеют жёсткой структуры, так как формировались пользователями вручную на протяжении нескольких лет. Чтобы решить эту проблему, была реализована идея импорта по JSON-сценарию. Сценарий – это JSON-документ [16], который фактически содержит указания для модуля импорта: откуда и какую информацию брать и куда и в каком виде размещать.

Как уже было отмечено, данные об испытаниях изделия хранятся в виде набора файлов различных форматов (MERA, текстовых, документах Excel), при этом размещение файлов задается специалистами

по обработке данных испытаний со значительными вариациями. Для проведения импорта данных для каждого испытанного изделия формируется JSON-файл, куда записывается метаданная обо всех постановках и запусках и пути к файлам трендов, коэффициентов калибровки, паспортов и т.д. Все параметры, описанные в этом файле, делятся на два типа – пользовательские и служебные.

Пользовательские параметры представляют собой реквизиты проведённых испытаний – дата, номер турбины и т.д. При импорте таких параметров не задействуется никакая-либо бизнес-логика; они просто сохраняются в базе и отображаются на интерфейсе системы в формате «ключ-значение».

К служебным параметрам относятся пути к необходимым файлам испытания, а также параметры навигации по файлу протокола испытания. Такие параметры система распознаёт и обрабатывает особым образом. Общая схема импорта представлена на рис. 2.

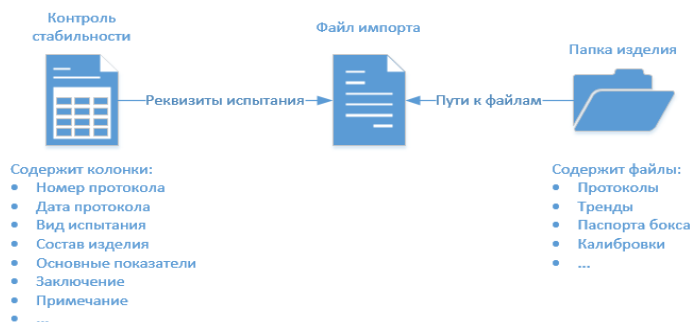


Рис. 2. Схема импорта при помощи JSON-файла импорта

```

{
  "1-й параметр стенда": "...",
  "2-й параметр стенда": "...",
  "Изделия": [
    {
      "1-й параметр изделия": "...",
      "2-й параметр изделия": "...",
      "Серии работ": [
        {
          "1-й параметр серии": "...",
          "2-й параметр серии": "...",
          "Работы": [
            {
              "1-й параметр работы": "...",
              "2-й параметр работы": ..."
            },
            {...}
          ]
        },
        {...}
      ]
    },
    {...}
  ]
}
    
```

Рис. 3. Иерархическая структура файла импорта

Чтобы импортировать все данные по испытанному изделию с учетом иерархии данных (изделие – серия работ – постанковка), в файле импорта создаётся иерархия узлов, отвечающих за каждый уровень испытания (рис. 3).

Набор параметров для каждой работы, как и иерархия работ и постановок в рамках испытания, не фиксирован. Для описания такой динамической структуры формат JSON подходит как нельзя лучше.

Кроме того, JSON-файл легко читается и пишется человеком, а также десериализуется в структуры данных современных языков программирования, что является несомненным плюсом при разработке программ для импорта. Использование json-модели для хранения результатов испытаний рассмотрено в [17].

8. Архитектура системы (экосистемы хранения данных испытаний). Вся системы сбора, обработки и хранения данных является комплексом различных приложений, которые вместе составляют так называемую экосистему хранения данных испытаний (рис. 4).

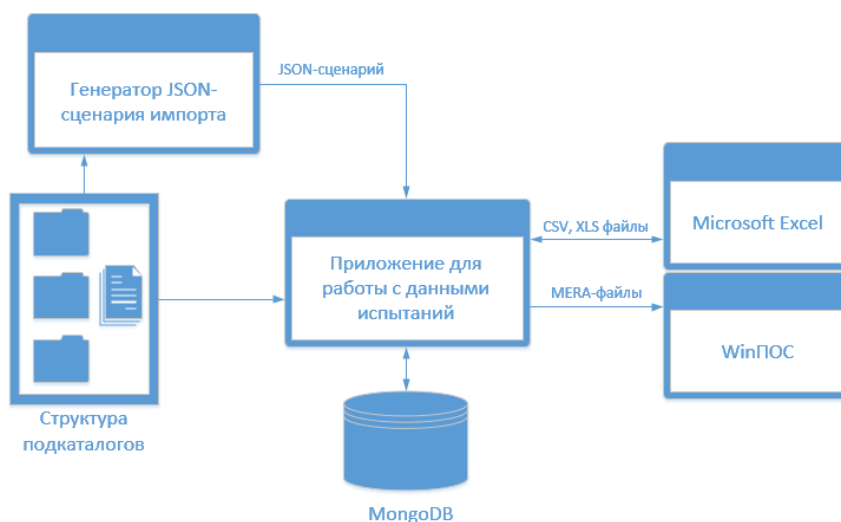


Рис. 4. Экосистема хранения и обработки постэкспериментальных данных

Основную часть функций системы покрывает программа для работы с архивными данными испытаний. Это центральный компонент системы, который содержит в себе сервисы работы с данными, расчётно-аналитический функционал и связывает остальные компоненты системы в единое целое с помощью импорта и экспорта данных.

9. Функции клиентского приложения. Данные всех испытаний, которые были импортированы в БД, представлены пользователям в виде иерархического дерева, отображающего исходную структуру испытаний (рис. 5). В левой части окна располагается «дерево» испытаний, в правой – параметры выделенного узла в «дереве» (они же реквизиты). С любым из узлов могут быть ассоциированы файлы, имеющие отношение к данному уровню испытаний.

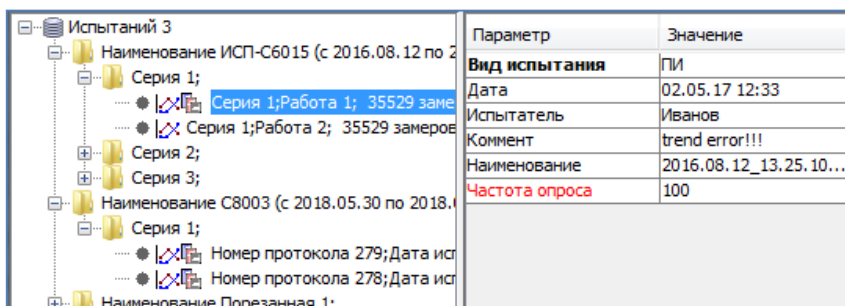


Рис. 5. Отображение дерева испытаний

Основная задача, стоящая перед специалистами, обрабатывающими информацию об испытаниях, – обеспечить поиск данных в трендах по заданным критериям. Рассмотрению задачи поиска данных во временных рядах посвящены следующие источники [18–20]. Для обнаружения особенностей во временных рядах в них используются методы, основанные на прогнозировании, при этом под объектами поиска, как правило, подразумеваются те или иные аномалии данных. Для анализа данных об испытаниях такие методы поиска не соответствуют требованиям, предъявляемым к способам анализа трендов. Специалисты по обработке данных должны иметь возможность непосредственно описывать объект поиска, задавая условия поискового запроса.

Из-за огромного объема хранящихся в базе данных поиск по всем трендам делать бесперспективно. Поэтому необходимо предварительно определить область поиска. Область может быть определена вручную выбором определенных наборов узлов, при этом расположение узлов соответствует исходной иерархии испытаний. Поиск будет производиться от выбранного узла по всем дочерним рекурсивно – вплоть до узла-запуска.

Другим способом является предварительный отбор узлов по параметрам узла или параметрам статического режима, тем самым про-

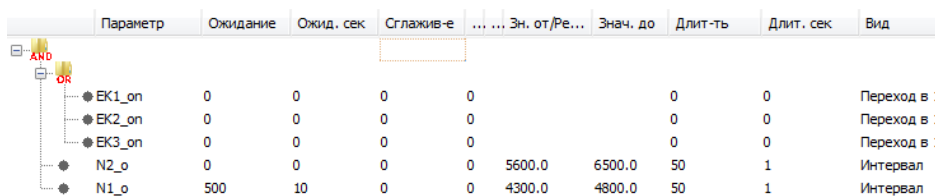
изводится отбор трендов, по которым будет осуществляться поиск. На каждом уровне заданной иерархии (Испытание, Серия, Работа) пользователю доступен поиск определенного набора параметров, заданного при импорте данных в БД. На уровне тренда параметры могут характеризовать как весь узел (тренд) в целом, так и отдельные режимы тренда. В зависимости от типа данных параметра для ввода условий поиска предусмотрены различные способы их ввода – задание диапазона искоемых значений или по совпадению (полному или частичному) строкового параметра.

Поиск особенностей в трендах сводится к поиску набора событий, характеризующих поведение тренда. В качестве этих событий рассматриваются такие, как попадание значения параметра (или его производной) в заданный диапазон, достижение минимума/максимума значения параметра, переключение состояния дискретного параметра и т.д. Каждое из перечисленных событий, в свою очередь, может описываться набором параметров. Так, событие типа «попадание в заданный диапазон» описывается заданием верхней и нижней границы диапазона, ожидаемым временем появления второго события после появления первого, в течение которого значение параметра не должно выходить за указанный интервал значений, размер окна усреднения при применении сглаживания параметра по алгоритму скользящего среднего (окна).

Искомые события могут происходить как одновременно, так и с временным сдвигом (с заданием времени ожидания, т.е. времени появления второго события после появления первого). В условиях поиска может быть задано несколько событий, при этом события могут группироваться при помощи логических операций И, ИЛИ. Для пользователя комбинирование условий при помощи И, ИЛИ представлено в виде дерева, которое также отображается в виде строки. Например, для условия поиска, который можно описать так: «найти моменты времени, когда во время режима «0,65» происходит переход параметра K2_close из 0 в 1 или из 1 в 0», строковое представление будет иметь вид:

(Режим=0,65) AND ((K2_close Переход в 1) OR (K2_close Переход в 0))

Для пользователя поисковый запрос имеет древовидную структуру, при этом сложность структуры запроса ограничена только возможностями специалиста в формулировании цели поиска. Пример дерева поиска особенностей представлен на рис. 6.



Параметр	Ожидание	Ожид. сек	Сглажив-е	...	Эн. от/Ре...	Знач. до	Длит-ть	Длит. сек	Вид
● EK1_on	0	0	0	0		0	0		Переход в 1
● EK2_on	0	0	0	0		0	0		Переход в 1
● EK3_on	0	0	0	0		0	0		Переход в 1
● N2_o	0	0	0	0	5600.0	6500.0	50	1	Интервал
● N1_o	500	10	0	0	4300.0	4800.0	50	1	Интервал

Рис. 6. Дерево поиска событий, характеризующих поведение тренда

Описание условий поиска составляет поисковый запрос. Поисковые запросы могут сохраняться пользователями для повторного использования. После получения результатов поиска возможна их дальнейшая обработка, также найденные тренды можно сразу просмотреть с помощью WinПОС.

Выводы. Разработанный комплекс приложений обеспечивает импорт информации об испытаниях, хранение и предоставление по запросу пользователей всей требуемой информации обо всех когда-либо проведенных испытаниях с возможностью экспорта в выбранный формат. Механизм поисковых запросов предоставляет специалистам по обработке данных мощный инструмент анализа.

Библиографический список

1. ГОСТ 16504-81. Система государственных испытаний продукции. Испытания и контроль качества продукции. Основные термины и определения (изд. май 2011 г. с изм. N 1) // Доступ из справ.-правовой системы КонсультантПлюс.
2. WinПОС после-экспериментальная обработка данных [Электронный ресурс]. – URL: <http://nppmega.ru/winpos>
3. Мишулина О.А. Статистический анализ и обработка временных рядов. – М.: Изд-во МИФИ, 2004. – С. 180.
4. Интеллектуальный анализ данных в управлении производственными системами (подходы и методы): монография / Л.А. Мыльников, Б. Краузе, М. Кютц, К. Баде, И.А. Шмидт. – М.: Библио-Глобус, 2017. – 334 с. DOI: 10.18334/9785950050176 (Гл. 5. Особенности работы с временными рядами).
5. Осипов Д.Л. Технологии проектирования баз данных. – М.: ДМК Пресс, 2019. – 498 с.
6. Зеленков Ю.А., Чувиллин В.Ю., Журавлев В.Е. Комплексная автоматизация испытаний газотурбинных двигателей. Ч. 2. Хранение

и обработка данных // Вестник Уфим. гос. авиацион. техн. ун-та. – 2011. – Т. 15. – № 2(42). – С. 126–131.

7. Asay Matt. Why time series databases are exploding in popularity. TechRepublic [Электронный ресурс]. – 26 June 2019. – URL: <https://www.techrepublic.com/article/why-time-series-databases-are-exploding-in-popularity/>

8. Как мы тестировали несколько баз данных временных рядов (1 августа 2019) [Электронный ресурс]. – URL: <https://habr.com/ru/company/itsumma/blog/462111>

9. Локотко А.В., Певзнер А.С., Яковлева Н.В. Программный комплекс сбора, обработки и хранения данных аэродинамического эксперимента // Информационные и математические технологии в науке и управлении. – 2017. – № 2(6). – С. 123–131.

10. Адаптивная архитектура программно-аппаратного комплекса хранения и обработки данных / Ю.А. Костиков, В.Ю. Павлов, А.М. Романенков, В.Б. Терновсков // Экономика: вчера, сегодня, завтра. – 2017. – Т. 7, № 9А. – С. 192–207.

11. Бэнкер Кайл. MongoDB в действии. – М.: ДМК Пресс, 2012. – 395 с.

12. Шмидт И.А., Петроченков А.Б., Даденков Д.А. Разработка системы хранения временных рядов в документо-ориентированной базе данных // Информационно-измерительные и управляющие системы. – 2019. – Т. 17, № 4. – С. 5–11.

13. Попов А.П., Шмидт И.А. Оптимизация хранения данных испытания сложных технических изделий в документо-ориентированной базе данных // Материалы междунар. конф. по мягким вычислениям и измерениям. – 2018. – Т. 1. – С. 169–172.

14. Васнев Н.В., Шмидт И.А. Система регистрации параметров испытаний сложных изделий на основе документно-ориентированной базы данных // Фундаментальные исследования. – 2016. – № 11–3. – С. 500–504.

15. Шмидт И.А., Попов А.П. Разработка оптимального метода хранения временных рядов в документоориентированной базе данных // Инновационные технологии: теория, инструменты, практика. – 2018. – Т. 1. – С. 233–238.

16. Введение в JSON [Электронный ресурс]. – URL: <http://json.org/json-ru.html>

17. Шмидт И.А., Васенев Н.В. Выбор оптимальной json-модели для хранения результатов испытаний // *Фундаментальные исследования*. – 2016. – № 11–3. – С. 620–625.

18. Chandola V., Banerjee A., Kumar V. Anomaly Detection A Survey // *ACM Computing Surveys*. – July 2009. – Vol. 41(3). – Article 15. – P. 1–72.

19. Шабельников А.Н., Шабельников В.А. Поиск аномалий в технических базах данных временных рядов // *Известия Южного федерал. ун-та. Технические науки*. – 2008. – Т. 81. – Вып. 4. – С. 167–173.

20. Антипов С.Г., Фомина М.В. Проблема обнаружения аномалий в наборах временных рядов // *Программные продукты и системы*. – 2012. – № 2. – С. 79–83.

References

1. GOST 16504-81. Sistema gosudarstvennykh ispytaniy produktsii. Ispytaniia i kontrol' kachestva produktsii. Osnovnye terminy i opredeleniia (izd. mai 2011 g. s izm. N 1) [State standard 16504-81. The state system of testing products. Product test and quality inspection. General terms and definitions (ed. May 2011 as amended No. 1)]. Dostup iz spravochno-pravovoi sistemy Konsul'tantPlius.

2. WinPOS posle-eksperimental'naia obrabotka dannykh [WinPOS post-experimental data processing], available at: <http://nppmera.ru/winpos>

3. Mishulina O.A. Statisticheskii analiz i obrabotka vremennykh riadov [Statistical analysis and processing of time series]. Moscow: Moskovskii inzhenerno-fizicheskii institut, 2004, 180 p.

4. Myl'nikov L.A., Krauze B., Kiutts M., Bade K., Shmidt I.A. Intel'ktual'nyi analiz dannykh v upravlenii proizvodstvennymi sistemami (podkhody i metody) [Intelligent data analysis in the management of production systems (approaches and methods) (Berichte aus der Wirtschaftsinformatik)]. Moscow: Biblio-Globus, 2017, 334 p. DOI: 10.18334/9785950050176 (Glava 5. Osobennosti raboty s vremennymi riadami).

5. Osipov D.L. Tekhnologii proektirovaniia baz dannykh [Database design Technologies]. Moscow: DMK Press, 2019, 498 p.

6. Zelenkov Iu.A., Chuvilin V.Iu., Zhuravlev V.E. Kompleksnaia avtomatizatsiia ispytaniy gazoturbinnnykh dvigatelei. Chast' 2. Khranenie i obrabotka dannykh [Comprehensive automation of testing of gas turbine engines. Part 2. Data storage and processing] *Vestnik Ufimskogo*

gosudarstvennogo aviatsionnogo tekhnicheskogo universiteta, 2011, vol. 15, no. 2(42), pp. 126-131.

7. Asay Matt. Why time series databases are exploding in popularity. TechRepublic. 26 June 2019, available at: <https://www.techrepublic.com/article/why-time-series-databases-are-exploding-in-popularity/>

8. Kak my testirovali neskol'ko baz dannykh vremennykh riadov (1 avgusta 2019) [How we tested several time series databases], available at: <https://habr.com/ru/company/itsumma/blog/462111>

9. Lokotko A.V., Pevzner A.S., Iakovleva N.V. Programmnyi kompleks sbora, obrabotki i khraneniia dannykh aerodinamicheskogo eksperimenta [Software package for collecting, processing and storing data of an aerodynamic experiment]. *Informatsionnye i matematicheskie tekhnologii v nauke i upravlenii*, 2017, no. 2(6), pp. 123-131.

10. Kostikov Iu.A., Pavlov V.Iu., Romanenkov A.M., Ternovskov V.B. Adaptivnaia arkhitektura programmno-apparatnogo kompleksa khraneniia i obrabotki dannykh [Adaptive architecture of hardware and software complex for data storage and processing]. *Ekonomika: vchera, segodnia, zavtra*, 2017, vol. 7, no. 9A, pp. 192-207.

11. Benker Kail. MongoDB v deistvii [MongoDB in Action]. Moscow: DMK Press, 2012, 395 p.

12. Shmidt I.A., Petrochenkov A.B., Dadenkov D.A. Razrabotka sistemy khraneniia vremennykh riadov v dokumento-orientirovannoi baze dannykh [Development of a time series storage system in a document-oriented database]. *Informatsionno-izmeritel'nye i upravliaiushchie sistemy*, 2019, vol. 17, no. 4, pp. 5-11.

13. Popov A.P., Shmidt I.A. Optimizatsiia khraneniia dannykh ispytaniia slozhnykh tekhnicheskikh izdelii v dokumento-orientirovannoi baze dannykh [Optimization of data storage for testing complex technical products in a document-oriented database] *Materialy mezhdunarodnoi konferentsii po miagkim vychisleniiam i izmereniiam*, 2018, vol. 1, pp. 169-172.

14. Vasnev N.V., Shmidt I.A. Sistema registratsii parametrov ispytaniia slozhnykh izdelii na osnove dokumentno-orientirovannoi bazy dannykh [System of registration of test parameters of complex products based on a document-oriented database]. *Fundamental'nye issledovaniia*, 2016, no. 11-3, pp. 500-504.

15. Shmidt I.A., Popov A.P. Razrabotka optimal'nogo metoda khraneniia vremennykh riadov v dokumentoorientirovannoi baze dannykh [Development of an optimal method for storing time series in a document-

oriented database]. *Innovatsionnye tekhnologii: teoriia, instrumenty, praktika*, 2018, vol. 1, pp. 233-238.

16. Vvedenie v JSON [Introduction to JSON], available at: <http://json.org/json-ru.html>

17. Shmidt I.A., Vasenev N.V. Vybor optimal'noi json-modeli dlia khraneniia rezul'tatov ispytaniia [Choosing the optimal json model for storing test results]. *Fundamental'nye issledovaniia*, 2016, no. 11-3, pp. 620-625.

18. Chandola V., Banerjee A., Kumar V. Anomaly Detection A Survey. *ACM Computing Surveys*, July 2009, vol. 41(3), art. 15, pp. 1-72.

19. Shabel'nikov A.N., Shabel'nikov V.A. Poisk anomalii v tekhnicheskikh bazakh dannykh vremennykh riadov [The search of anomalies in the technical bases of time series data]. *Izvestiia Iuzhnogo federal'nogo universiteta. Tekhnicheskie nauki*, 2008, vol. 81, iss. 4, pp. 167-173.

20. Antipov S.G., Fomina M.V. Problema obnaruzheniia anomalii v naborakh vremennykh riadov [The problem of detecting anomalies in time series sets]. *Programmnye produkty i sistema*, 2012, no. 2, pp. 79-83.

Сведения об авторах

Шмидт Игорь Альбертович (Пермь, Россия) – кандидат технических наук, доцент кафедры «Микропроцессорные средства автоматизации» Пермского национального исследовательского политехнического университета (614990, Пермь, Комсомольский пр., 29, e-mail: shmidt-ia@yandex.ru).

Желтышев Артем Сергеевич (Пермь, Россия) – студент Пермского национального исследовательского политехнического университета (614990, Пермь, Комсомольский пр., 29, e-mail: artem.zheltyshev.98@mail.ru).

About the authors

Schmidt Igor Albertovich (Perm, Russian Federation) is a Ph. D. in Technical Sciences, Associate Professor of the Department of Microprocessor automation tools Perm National Research Polytechnic University (614990, Perm, 29, Komsomolsky pr., e-mail: shmidt-ia@yandex.ru).

Zheltyshev Artem Sergeevich (Perm, Russian Federation) is a Student Perm National Research Polytechnic University (614990, Perm, 29, Komsomolsky pr., e-mail: artem.zheltyshev.98@mail.ru).

Получено 17.08.2020