

Министерство образования и науки Российской Федерации

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ  
ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ  
«САРАТОВСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ  
ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ ИМЕНИ Н. Г. ЧЕРНЫШЕВСКОГО»

Кафедра математической  
кибернетики и компьютерных наук

**РАЗРАБОТКА ИНФОРМАЦИОННОЙ СИСТЕМЫ ДЛЯ ПОИСКА  
КРИТИЧЕСКИХ СИТУАЦИЙ В ПРОЦЕССЕ ЭКСПЛУАТАЦИИ  
ИСТРЕБИТЕЛЯ RAFALE**

АВТОРЕФЕРАТ МАГИСТЕРСКОЙ РАБОТЫ

студента 2 курса 271 группы  
направления 09.04.01 — Информатика и вычислительная техника  
факультета КНиИТ  
Плодистого Петра Дмитриевича

Научный руководитель  
профессор, д.т. н.

\_\_\_\_\_

В. А. Кушников

Заведующий кафедрой  
к. ф.-м. н.

\_\_\_\_\_

С. В. Миронов

Саратов 2018

## СОДЕРЖАНИЕ

ВВЕДЕНИЕ .....	3
1 Раздел 1. Модели, методы и алгоритмы используемые при определении критических ситуаций в процессе эксплуатации истребителя Rafale .....	5
2 Раздел 2. Выбор и обоснование инструментов и средств для разработки информационной системы .....	7
2.0.1 MySQL .....	8
2.0.2 JS .....	9
2.0.3 Materialize.css .....	9
2.0.4 Treant.js .....	9
3 Раздел 3. Разработка графического интерфейса пользователя информационной системы .....	10
ЗАКЛЮЧЕНИЕ .....	13
СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ .....	14

## ВВЕДЕНИЕ

Темой работы является разработка информационной системы для поиска критических ситуаций в процессе эксплуатации истребителя RAFALE, данная тема актуальна в связи с тем необходимостью решения задачи обеспечения и поддержания безопасности и предотвращения критических ситуаций, возникающих в процессе функционирования авиационно-транспортных систем.

Несмотря на совершенствование существующих систем, количество аварий, происшествий и их опасность не снижается. Необходимо также отметить, что полностью безопасных систем не существует, и фактически все сложные системы функционируют при неопасных, не приводящих к авариям, ошибках персонала и неисправностях техники [1–3]. Современные сложные технические комплексы состоят из значительного количества частей, внутри них протекают процессы различной природы, поэтому для эффективного исследования аварий необходимо совершенствование не только средств диагностики состояния системы, но и моделей, методов и алгоритмов, применяемых для анализа развития и функционирования этих систем.

Задача по обеспечению безаварийного и безопасного функционирования авиационно-транспортных систем становится все более актуальной. Как свидетельствуют данные о многочисленных авариях, многие из них могли бы быть предотвращены путем своевременной комплексной оценки ресурсов и возможностей подсистем в совокупности с ресурсами персонала. Традиционно используемое понятие остаточного ресурса не дает достаточной точности в оценке безопасности. При расчете остаточного ресурса как предполагаемой суммарной наработки системы от момента контроля до перехода в предельное состояние не учитывается, как правило, ряд сложных взаимодействий процессов в системе, в особенности человеко-машинное взаимодействие. В то же время большая часть аварий и катастроф происходит по причине бесконтрольных действий персонала управления, и в последние десятилетия эта доля растет. С другой стороны, во многих отраслях наблюдается значительный износ парка техники и оцениваемый остаточный ресурс машин приближается к тем граничным значениям, при которых использование его величины при расчетах безопасного функционирования на достаточно длительных промежутках времени связано со значительным риском аварии.

В настоящее время разработаны и прошли проверку практикой сложные

системы управления АТС, разработаны математические модели, методы и алгоритмы поиска неисправностей авиационной техники в том числе при неблагоприятном сочетании событий. При этом широко использовались системно-динамический подход и модели системной динамики [4, 5]. Кроме того, в системах управления АТС часто используются алгоритмы построения деревьев событий, позволяющие на различных временных интервалах диагностировать возникающие неисправности.

Для уменьшения числа авиационных происшествий можно использовать информационные системы, позволяющие анализировать и систематизировать причины возникновения критических ситуаций на основе данных полученных в процессе эксплуатации, проектирования, тестов и т.д.

Целью данной работы является разработка информационной системы для поиска критических ситуаций в процессе эксплуатации истребителя Rafale. Были поставлены следующие задачи:

- изучить модели и методы для определения критических ситуаций;
- выбрать из них наиболее подходящие для истребителя;
- на основе изученных методов разработать алгоритм для определения критических ситуаций в истребителе Rafale;
- разработать информационную систему для поиска критических ситуаций в процессе эксплуатации истребителя Rafale.

Дипломная работа представлена введением, тремя разделами, заключением, списком использованных источников и приложением (листингом программы).

Раздел 1 представляет собой описание моделей, методов и алгоритмов при определении критических ситуаций в процессе эксплуатации истребителя Rafale.

Раздел 2 представляет собой описание инструментов и средств для разработки информационной системы

Раздел 3 представляет собой описание разработки графического интерфейса пользователя информационной системы

# 1 Раздел 1. Модели, методы и алгоритмы используемые при определении критических ситуаций в процессе эксплуатации истребителя Rafale

Данный раздел состоит из трех частей. В первой описываются модели и методы поиска критических ситуаций. Такие как метод исследования опасностей и функционирования (HAZOP), анализа видов и последствий отказов (failure mode and effects analysis – FMEA) и анализа дерева неисправности (faulttreeanalysis – FTA). В этой части приводится их характеристики, а так же их сильные и слабые стороны.

Во второй части работы рассматриваются процедуры построения комплексов причинно-следственных связей(для событий,цепочек событий, деревьев), использованных при анализе критических сочетаний в процессе эксплуатации самолета Dassault Rafale. Так же приведена иерархическая схема самолета Dassault Rafale. В третьей части представлен алгоритм для выявления и предотвращения критических ситуаций для самолета Dassault Rafale.

Для решения задачи предлагается осуществить следующие шаги:

Шаг 1. Построение схемы самолета и возможных отказов и разработка алгоритмов решения задачи их предотвращения. Определить множество  $\bar{A} = A_1, A_2, \dots, A_n$  критических ситуаций.

$$P_i(\lambda_1(t), \lambda_2(t), \dots, \lambda_k(t), \mu_1(t), \mu_2(t), \dots, \mu_k(t), x(t), t), i = 1, \dots, n \quad (1)$$

$$F_i(\lambda_1(t_H), \lambda_2(t_H), \dots, \lambda_k(t_H), \mu_1(t_H), \mu_2(t_H), \dots, \mu_k(t_H), x(t_H), t_H) = F_i^H, \quad (2)$$

$$F_i(\lambda_1(t_K), \lambda_2(t_K), \dots, \lambda_k(t_K), \mu_1(t_K), \mu_2(t_K), \dots, \mu_k(t_K), x(t_K), t_K) = F_i^K, \quad (3)$$

$$i = 1, \dots, n. \quad (4)$$

$$C_j \leq G_j(\lambda_1(t), \lambda_2(t), \dots, \lambda_k(t), \mu_1(t), \mu_2(t), \dots, \mu_k(t), x(t), t) \leq D_j, \\ j = 1, \dots, m, \quad (5)$$

- Построить множество деревьев отказов  $\bar{D} = D_1, D_2, \dots, D_n$ , соответствующих ситуациям из множества с множеством  $\bar{A}$  элементарных событий  $E = e_1, \dots, e_k$ .
- Для каждого  $e_i, i = 1, \dots, k$  определить комплекс инструкций  $Q(\mu_i(t))$

обеспечивающий парирование этого события с интенсивностью  $\mu_i(t)$  и соответствие между значениями интенсивности парирования  $\mu_i(t)$  события  $e_i$  и списками действий  $Q(\mu_i(t))$  для обеспечения этой интенсивности.

- Для каждого дерева отказов  $D \in \overline{D}$  определить все его минимальные сечения.
- Разработать комплекс алгоритмов для решения поставленной задачи определения вероятностей

$$P_i(\lambda_1(t), \lambda_2(t), \dots, \lambda_k(t), \mu_1(t), \mu_2(t), \dots, \mu_k(t), x(t), t) \leq \varepsilon_i(t), i = 1, \dots, n$$

возникновения отказов и определения интенсивностей  $\mu_1(t), \mu_2(t), \dots, \mu_k(t)$  их восстановления и парирования событий из множества  $E = e_1, \dots, e_k$ , при которых для заданных интенсивностей возникновения этих событий  $\lambda_1(t), \lambda_2(t), \dots, \lambda_k(t)$  и воздействий среды  $x(t)$  достигает максимума критерий 1 с ограничениями 2 и граничными условиями 5.

Шаг 2. Моделирование критических сочетаний событий.

- Построение комплексов для моделирования аварийных ситуаций на различных временных интервалах.
- Моделирование аварийных ситуаций и уточнение вероятности их возникновения.
- Занесение информации в базе данных и корректировка модели по результатам моделирования.
- Разработка рекомендация по предотвращению аварийных ситуаций на различных временных интервалах и этапах развития.

Шаг 3. Внедрение разработанной интеллектуальной системы поддержки решений в информационные системы.

## **2 Раздел 2. Выбор и обоснование инструментов и средств для разработки информационной системы**

Данный раздел состоит из двух частей в первой производится постановка задачи и разработка бизнес-правил. Для начала необходимо понять как будет использоваться разрабатываемая система. На этапе проектирования конструкторское бюро передает сведения для заполнения системы и уже на этом этапе система определяет ошибки проектирования, которые могут привести к выходу самолета из строя. Система создает отчет для конструкторского бюро с указанием ошибок проектирования ( если они имеются). После этого самолет поступает в эксплуатацию. При возникновении происшествий с самолетом, в дело вступает специальная комиссия, которая производит расследование [6]. Данные, полученные в процессе расследования передаются в систему и конструкторское бюро, для корректирования модели, затем системой создается отчет, который отправляется в конструкторское бюро и в агентство по расследованию происшествий.

Формализованное описание задачи в нашем для системы будет выглядеть следующим образом.

Наименование задачи:

Создание информационной системы.

Цель работы системы:

предотвращении критических ситуаций при эксплуатации самолета.

Функции системы:

- получение данных от КБ;
- составление отчетов;
- получение данных от комиссий при авиационных происшествиях;
- анализ данных;
- редактирование модели;

Требования к программе:

Компьютер с веб-браузером и доступом к сети Интернет

Перечень вводимой информации:

- название детали или узла самолета;
- артикул;
- производитель детали или узла самолета;
- конструктор детали или узла самолета;

- описание детали или узла самолета;
- характеристики детали или узла самолета;
- данные об испытаниях детали или узла самолета;
- вероятность выхода из строя детали или узла самолета;
- вероятность предотвращения последствий выхода из строя детали или узла самолета;

Перечень печатных отчетов:

- отчет о критических элементах самолета;
- отчет о вероятности выхода из строя самолета;
- отчет о причине выхода из строя самолета;

Реализацией информационной системы, является web-приложение.

Во второй части описываются средства при помощи которых разрабатывается система.

Go – является многопоточным компилируемым языком программирования, которой разработала компания Google. Разработка языка началась осенью 2007 года. Проектированием и разработкой занимались Роберт Гризмер, Роб Пайк и Кен Томпсон. Презентация языка состоялась осенью 2009 года. В настоящий момент существуют компиляторы для операционных систем OpenBSD, FreeBSD, Linux, OS X, Mac, Windows. В версии языка 1.3 добавлена поддержка Solaris, Plan 9 и DragonFly BSD, а в 1.4 была реализована поддержка платформы Android. В данный момент язык Go является активно развивающимся, для него разрабатывается множество библиотек и фреймворков.

Go используется, в основном, для написания микросервисов и серверных приложений. Он сочетает в себе высокую скорость и кроссплатформенность.

Этот язык был выбран по причине его кроссплатформенности, то есть приложение можно запустить как на операционной системе linux, так и на windows и macOS.

Язык продолжает развиваться, и разработчики рассматривают возможность включения в язык средств обобщённого программирования. В «Часто задаваемых вопросах» по языку приводятся аргументы против использования утверждений, а наследование без указания типа, наоборот, отстаивается.

## 2.0.1 MySQL

MySQL - реляционная система управления базами данных. MySQL поддерживает SQL (структурированный язык запросов) и может применяться в

качестве SQL-сервера. Это означает, что общаться с сервером можно на языке SQL: клиент посыпает серверу запрос, тот его обрабатывает и отдает клиенту только те данные, которые были получены в результате этого запроса [7]. Тем самым клиенту не требуется выкачивать данные и производить вычисления. Преимущества MySQL:

### 2.0.2 JS

JavaScript – мультипарадигменный язык программирования. Поддерживает объектно-ориентированный, императивный и функциональный стили. Является реализацией языка ECMAScript.

JavaScript обычно используется как встраиваемый язык для программного доступа к объектам приложений. Наиболее широкое применение находит в браузерах как язык сценариев для придания интерактивности веб-страницам.

JavaScript является объектно-ориентированным языком, но используемое в языке прототипирование обуславливает отличия в работе с объектами по сравнению с традиционными класс-ориентированными языками [8]. Кроме того, JavaScript имеет ряд свойств, присущих функциональным языкам — функции как объекты первого класса, объекты как списки, карринг, анонимные функции, замыкания — что придаёт языку дополнительную гибкость [9].

### 2.0.3 Materialize.css

В этом проекте также используется css библиотека materialize.css, которая позволяет уделять меньше внимания разметке и больше – функциональности приложения. Также использование данной библиотеки позволяет сделать красивый и удобный интерфейс с точки зрения пользователя приложения.

### 2.0.4 Treant.js

Treant.js это библиотека для JS для создания деревьев, используя HTML, CSS и SVG. С помощью неё можно построить дерево, основываясь на добавленных элементах самолёта.

### **3 Раздел 3. Разработка графического интерфейса пользователя информационной системы**

В данном разделе описывается разработанная система.

Данное приложение состоит из 12 пакетов. Структура пакетов приложения выглядит следующим образом:

- main - основной пакет, который используется для запуска приложения и обработки главной страницы, страницы авторизации и регистрации. В этом пакете обозначаются ссылки и хандлеры, которые будут использоваться для обработки этих ссылок. Также, в этом пакете прописаны флаги командной строки, в которые можно написать путь до конфигурационного файла и порт, на котором будет слушать веб сервер.
  - detail - пакет, в котором происходит обработка действия с элементами авионики, фюзеляжа, двигателя и гидравлики, такие как создание нового элемента, редактирование и отображение.
  - plane - пакет, который содержит общие функции для работы с деталями самолёта, такие как поиск, добавление характеристик и тестов, получение информации о детали и получение всех деталей. Эти функции используются в других пакетах.
  - server - пакет, который обеспечивает загрузку конфигурационного файла, соединение с сервером баз данных и хранилище сессий.
  - templates - пакет с шаблонами для встраивания данных в страницы. Для их компиляции необходимо использовать библиотеку quicktemplates.
  - user - пакет для работы с пользователями: создание пользователей, проверка введённых данных авторизации, получение аватара с сервиса gravatar, хеширование пароля и создание соли.
  - admin - интерфейс для управления пользователями, который вызывает функции из пакета user и отправляет результат пользователю
  - archive - пакет для архивирования
  - documents - пакет для создания отчётов и работы с ними
  - help - справка по системе
  - logs - для записи и отображения логов пользовательских действий
  - graph - пакет для обработки данных из базы данных и отображения дерева поломок
- Все вышеуказанные проекты объединяются в файле main.go.

В качестве базы данных используется MySQL версии 5.7.22.

База данных состоит из 7 таблиц, связанных между собой внешними ключами. Список таблиц базы данных и описание полей предоставлено ниже:

- documents - таблица для хранения созданных документов, которая содержит:
  - уникальный идентификатор документа
  - имя пользователя, создавшего отчёт
  - дата создания отчёта
- users - таблица для хранения пользователей, которая содержит:
  - уникальный идентификатор пользователя
  - имя пользователя для авторизации
  - хэш пароля, так как хранить пароль в открытом виде небезопасно
  - соль для хеширования пароля для дополнительной безопасности
  - почта для отправки уведомлений
  - права доступа к приложению
- type - таблица для хранения типа детали, которая содержит:
  - уникальный идентификатор типа
  - имя типа элемента (двигатель, фюзеляж, авионика, гидравлика)
- logs - таблица логов, которая содержит:
  - уникальный идентификатор записи
  - тип действия
  - имя пользователя, совершившего действие
  - элемент, с которым производилось действие
  - дата действия
- data - таблица данных о деталях самолёта, которая содержит:
  - уникальный идентификатор записи
  - наименование детали
  - предыдущий элемент
  - операция на предыдущий элемент
  - тип детали
  - код производителя
  - комментарий
  - описание
  - конструктор

- производитель
- ссылка на изображение
- дата добавления
- дата создания детали
- characteristics - таблица характеристик деталей самолёта, которая содержит:
  - уникальный идентификатор записи
  - наименование характеристики
  - значение
  - ссылка на идентификатор детали
- tests - таблица тестов для деталей самолёта, которая содержит:
  - уникальный идентификатор записи
  - дата теста
  - результат
  - ссылка на идентификатор детали

## ЗАКЛЮЧЕНИЕ

В результате выполнения научно-исследовательской работы по теме «разработка интерфейса информационно-справочной системы для выявления и предотвращения критических ситуаций (например управления истребителем Rafale)» получены следующие результаты:

- Были рассмотрены способы построения моделей и методов поиска критических ситуаций в сложных авиационных системах, такие как метод исследование опасностей и функционирования (HAZOP), метод анализа видов и последствий отказов (failure mode and effects analysis – FMEA) и метод анализа дерева неисправности (fault tree analysis – FTA)
- Предложен алгоритм для выявления и предотвращения критических ситуаций для самолета Dassault Rafale
- Разработана компоновочная схема для анализа надежности и безопасности функционирования истребителя Dassault Rafale
- Осуществлена постановка задачи, определены основные бизнес процессы подлежащие автоматизации, построены ER-диаграммы, что позволило создать основу разрабатываемой информационной системы, определить структуру клиент-серверного приложения и состав рабочих станций, обосновать систему триггеров безопасности данных.
- Предложена и обоснована структура базы данных (БД), определены первичные и вторичные ключи БД, выполнена нормализация БД, в результате проведения которой БД была приведена к третьей нормальной форме.
- Исходя из анализа графического интерфейса пользователя, разработана иерархическая система меню, предложены экранные формы ввода информации, составлена и апробирована процедура авторизации пользователя позволяющая по части его пароля отнести к одной из трех групп пользователей.
- Выбраны состав и структура комплекса технических средств, позволяющих осуществить аппаратную реализацию, разрабатываемой информационно-программного обеспечения.
- Разработана информационная система для поиска критических ситуаций в процессе эксплуатации истребителя Rafale.

## СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

- 1 *Резчиков, А. Ф.* Причинно-следственные комплексы взаимодействий производственных процессов. / А. Ф. Резчиков, В. А. Твердохлебов // *Проблемы управления*. — 2010. — № 3. — С. 51–59.
- 2 *Новожилов, Г. В.* Безопасность полета самолета: концепция и технология. / Г. В. Новожилов, М. С. Неймарк, Л. Г. Цесарский. — Москва: МАИ, 2007.
- 3 *Новожилов, Г. В.* Проблемы безопасности авиационно-транспортных и других сложных человеко-машинных систем / Г. В. Новожилов, А. Ф. Резчиков, М. С. Неймарк. — ООО Наука, 2012.
- 4 *Аветисян, Ю. А.* Математические модели и алгоритмы оперативного управления процессами ликвидации чрезвычайных ситуаций. / Ю. А. Аветисян, В. А. Кушников, А. Ф. Резчиков, В. А. Родичев // *Мехатроника, автоматизация, управление*. — 2009. — № 11. — С. 43–47.
- 5 Управление авиационно-транспортными системами на основе причинно-следственных деревьев событий / Г. В. Новожилов, А. Ф. Резчиков, М. С. Неймарк, Л. Г. Цесарский, В. А. Кушников, А. С. Богомолов, Л. Ю. Филимонюк, К. И. Шоломов // *Полет. Общероссийский научно-технический журнал*. — 2015. — № 6. — С. 13–17.
- 6 *Григорьев, М.* Проектирование информационных систем. Учебное пособие / М. Григорьев, И. Григорьева. — Москва: Юрайт, 2016.
- 7 Разработка программного комплекса с целью оптимизации способа хранения данных об измерениях счетчиков [Электронный ресурс]. — URL: <https://bibliofond.ru/view.aspx?id=868811> (Дата обращения 1.11.2017). Загл. с экрана. Яз. рус.
- 8 *Затонский, А.* Информационные технологии. Разработка информационных моделей и систем. Учебное пособие / А. Затонский. — Москва: Инфра-М, 2014.
- 9 *Хавербек, М.* Выразительный JavaScript / М. Хавербек. — 2015.