

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ
ВЫСШЕГО ОБРАЗОВАНИЯ
«САНКТ-ПЕТЕРБУРГСКИЙ ПОЛИТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ ПЕТРА ВЕЛИКОГО»
ВШ программной инженерии



ПОЛИТЕХ

Санкт-Петербургский
политехнический университет
Петра Великого

КУРСОВАЯ РАБОТА

**Разработка программного средства по обслуживанию
воздушного движения**
по дисциплине «Проектирование интеллектуальных систем управления»

Выполнила
Студент группы 3530202/70201

К.А. Рогова

Руководитель
доцент ВШ КФСУ, к.т.н.

В.А. Онуфриев

Санкт-Петербург
2020 г.

**ЗАДАНИЕ
НА ВЫПОЛНЕНИЕ КУРСОВОЙ РАБОТЫ**

студенту группы 3530202/70201 Роговой Ксении Анатольевне

1. Тема работы: разработка программного средства по обслуживанию воздушного движения.

2. Срок сдачи студентом законченной работы: 14.12.2020

3. Исходные данные к работе: должно быть разработано программное средство, составляющее расписание и очередность взлета и посадки воздушных судов в условиях неопределенности ситуации, основываясь на описании требований, ограничений и принципа работы программы для обслуживания взлетно-посадочных действий.

4. Содержание пояснительной записки: введение, постановка задачи, актуальность, разработка структуры программного средства, реализация модуля обучения, реализация программного средства, заключение, список использованных источников, приложения.

Примерный объем пояснительной записки: 30 страниц машинописного текста.

5. Перечень графического материала: IDEF0-диаграмма работы системы, функциональная схема.

6. Консультанты _____

7. Дата получения задания: «___» _____ 20__ г.

Руководитель, доц. ВШ КФСУ, к.т.н.

В. А. Онуфриев

Задание принял к исполнению
студент гр. 3530202/70201



К. А. Рогова

Оглавление

Введение.....	4
Постановка задачи.....	5
Актуальность.....	6
Разработка структуры программного средства.....	7
Реализация модуля обучения.....	9
Реализация программного средства.....	10
Заключение.....	12
Приложение.....	13

Введение

Интеллектуальная система (ИС) — это техническая или программная система, способная решать задачи, традиционно считающиеся творческими, принадлежащие конкретной предметной области, знания о которой хранятся в памяти такой системы. Структура интеллектуальной системы включает базу знаний и механизм вывода решений.

В данной работе разрабатывается интеллектуальная система по обслуживанию воздушного движения, которая способна в условиях неопределенности ситуации принимать решение о порядке посадки воздушных судов, основываясь на базе знаний и предыдущем опыте.

Постановка задачи

Необходимо разработать программное средство, составляющее расписание и очередность посадки воздушных судов в условиях неопределенности ситуации, основываясь на описании требований, ограничений и принципа работы программы для обслуживания взлетно-посадочных действий.

На вход ИС будет подаваться информация о самолетах, которые необходимо обслужить. Данные не всегда точны и могут содержать ошибки. Это приводит к неопределенности ситуации — на вход программы может прийти неопознанный объект, который ИС должна правильно обслужить, основываясь на знаниях и предыдущем опыте решения задач. У самолета есть приоритет, процент оставшегося топлива и состояние аварийности, а также модель и название.

На выходе ИС должна выводить вектор — очередность посадки с сопоставлением «борт — полоса». Каждый борт должен быть обслужен, каждая используемая полоса должна быть доступной. Очередность выстраивается на базе анализа приоритета и состояния судна.

В базе данных системы хранится информация о взлетных полосах и погодных условиях. Погодные условия прогнозируются нейронной сетью.

Актуальность

Диспетчерское обслуживание воздушного движения – одна из важнейших частей авиации. Управление воздушным движением позволяет предотвращать столкновения между воздушными судами и объектами на площади маневрирования, а также ускоряет и регулирует движение.

Сейчас сложную задачу обслуживания воздушного движения решают авиадиспетчеры, проходящие специальное обучение и подготовку. Нетривиальность задачи требует быстрых, точных и правильных решений, способных быть принятыми специалистами с большим опытом.

Интеллектуальная система, способная анализировать воздушный трафик и подходить к решению обслуживания движения, может снизить нагрузку на авиадиспетчера и позволит увеличить эффективность управления.

Разработка структуры программного средства

Структуру системы и описание происходящих процессов опишем с помощью функциональной модели и нотации IDEF0.

Система имеет базу знаний, базу данных, хранилища алгоритмов, формул и нейросетей. Компоненты, обращаясь к базам и хранилищам, решают отдельные подзадачи для получения конечного результата. Функции каждого из компонентов будут рассмотрены подробнее в главе «Реализация программного средства»

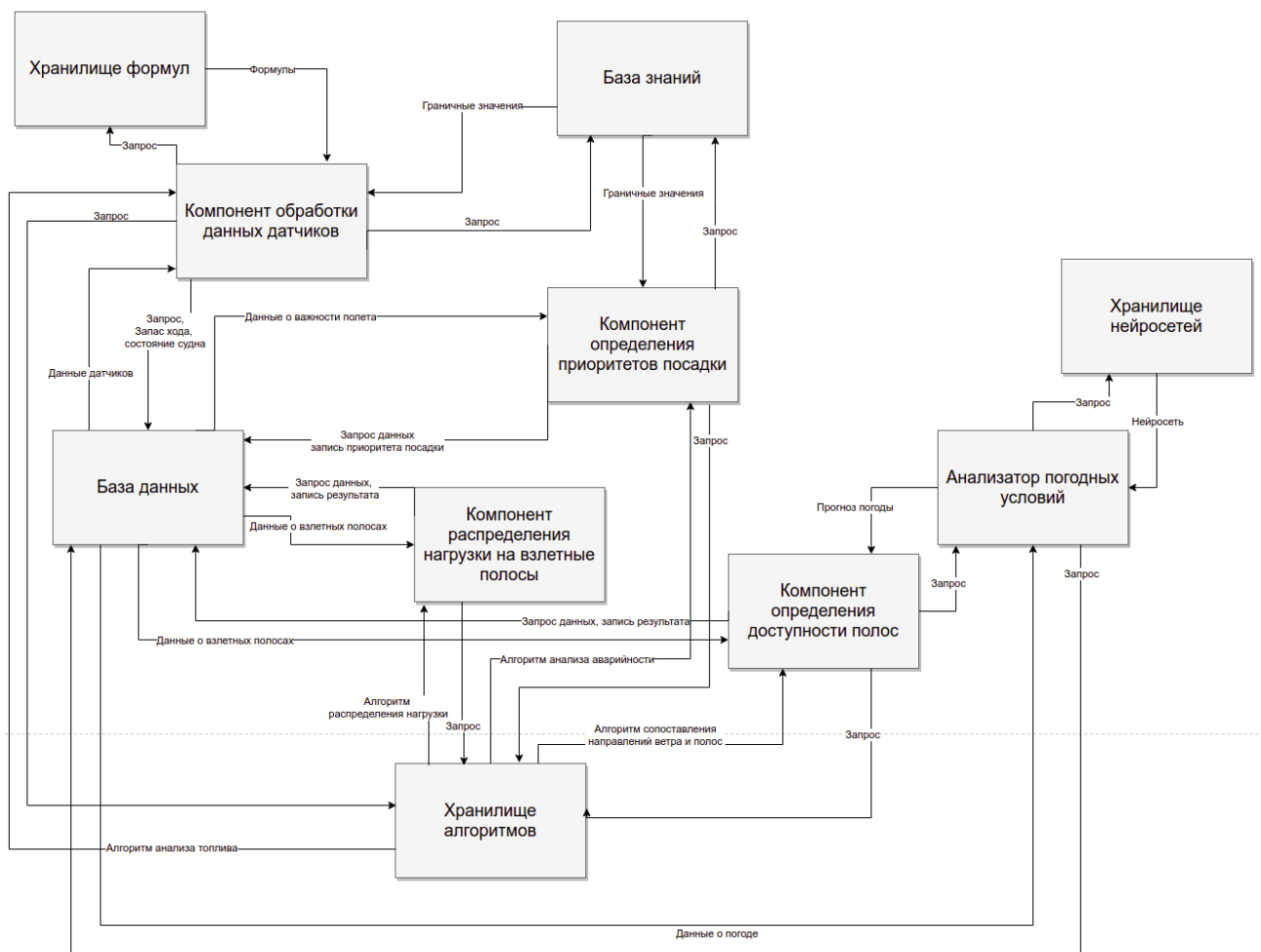


Иллюстрация 1: Функциональная модель

Описание последовательности процессов представлены на модели IDEF0, на ней можно наблюдать логическое отношение между компонентами.

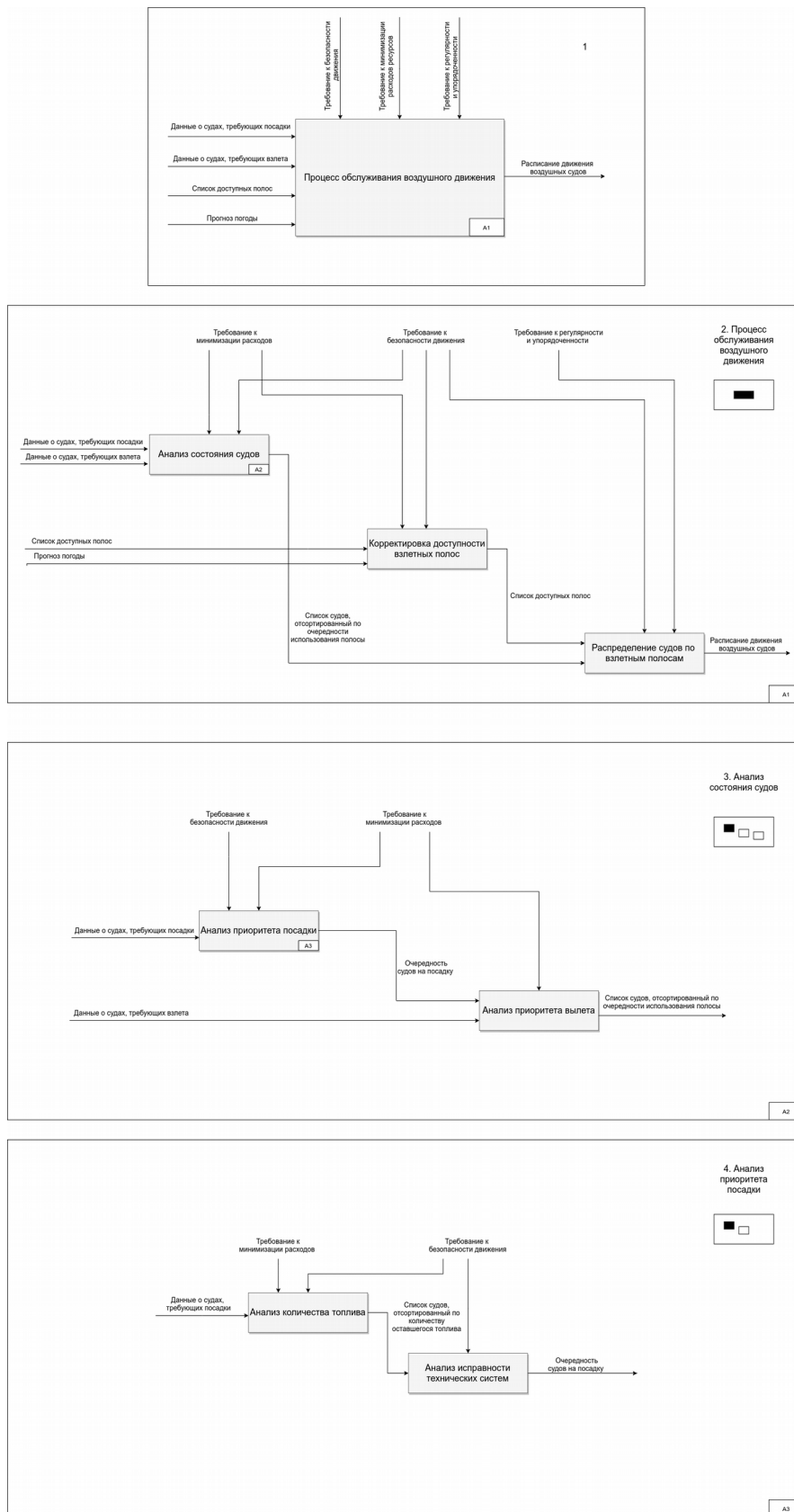


Иллюстрация 2: IDEF0

Реализация модуля обучения

Интеллектуальная система должна проанализировать входные данные для начала работы. Предполагается, что данные о самолетах могут быть неверны.

В базе знаний хранятся суждения о том, какие данные о самолете и какие ограничения для этих данных необходимы для корректного управления воздушным движением. ИС проверяет каждый борт из входных данных на соответствие данным правилам. Если ИС понимает, что какие-то данные неверны или отсутствуют, она попытается исправить эти значения на основе своего предыдущего опыта.

По принципу обучения без учителя ИС пытается самостоятельно найти корреляции в полях данных воздушного судна. Система находит общие признаки и проводит кластеризацию данных. Она предполагает, к какому из кластеров будет принадлежать неопознанный летающий объект, и заполняет недостающие или неправильные поля средними значениями данного кластера (значения кластерного центра).

Реализация программного средства

База знаний представлена в виде триплетов в TurtleNotation и хранится в файле knowledge_base.n3. Общение с базой происходит с помощью SPARQL-запросов. Реализованные триплеты:

```
mc:Formulas a rdfs:Class.
mo:formula a mc:Formulas.
mo:emergencyFormula a mc:Formulas;
    mp:checks mo:emergency.

mc:Algorithms a rdfs:Class.
mo:algorithm a mc:Algorithms.
mo:analyzeDirections a mc:Algorithms.
mo:analyze_fuel a mc:Algorithms;
    mp:checks mo:fuel_percent.

mc:NeuralNetwork a rdfs:Class.
mo:neural_network a mc:NeuralNetwork.
mo:weather_nn a mc:NeuralNetwork.

mc:Action a rdfs:Class.
mo:check_emergency a mc:Action;
    mp:needs mo:algorithm;
    mp:calls mo:analyze_emergency.
mo:check_fuel a mc:Action;
    mp:needs mo:algorithm;
    mp:calls mo:analyze_fuel.
mo:get_wind_info a mc:Action;
    mp:needs mo:neural_network;
    mp:calls mo:weather_nn.
mo:match_availability a mc:Action;
    mp:needs mo:algorithm;
    mp:calls mo:analyze_directions.
mo:get_order a mc:Action;
    mp:needs mo:algorithm;
    mp:calls mo:analyze_order.

mc:Aircrafts a rdfs:Class.
mc:Runways a rdfs:Class.

mc:Models a rdfs:Class.
mc:Priorities a rdfs:Class.
mc:Emergencies a rdfs:Class.
mc:Directions a rdfs:Class.
mc:Fuel a rdfs:Class.
mc:Time a rdfs:Class.
mc:Order a rdfs:Class.

mo:model a mc:Models.
mo:priority a mc:Priorities.
mo:emergency a mc:Emergencies;
    mp:priority_restrict_value 2.
mo:direction a mc:Directions.
mo:fuel_percent a mc:Fuel;
    mp:no_less_than 0;
    mp:no_more_than 100;
    mp:critical_value 5.

mo:airtime a mc:Time.

mo:aircraft a mc:Aircrafts;
    mp:contains mo:model, mo:fuel_percent, mo:priority, mo:emergency, mo:airtime;
    mp:calls mo:check_fuel, mo:check_emergency.
```

```
mo:runway a mc:Runways;  
  mp:contains mo:state, mo:direction;  
  mp:needs mo:get_wind_info;  
  mp:calls mo:match_availability.  
  
mo:order a mc:Order;  
  mp:calls mo:get_order.
```

Базы данных представлены в виде json- и csv- файлов. В них хранятся данные о взлетных полосах, погоде и уже обслуженных самолетах.

Хранилище нейронных сетей содержит в себе сеть для прогнозирования погоды. Модель обучается на готовом датасете и возвращает прогнозы о температуре, силе и направлении ветра.

Хранилище алгоритмов содержит в себе алгоритмы для анализа количества топлива, анализа аварийной ситуации, доступности взлетной полосы и правильного порядка вылета.

Компонент обработки данных датчиков, получая входные данные о самолетах, обращается к базе знаний, базе данных и алгоритмам и корректирует поля данных о суднах (если это необходимо), с помощью кластеризации определяя средние значения.

Компонент определения приоритетов посадки на основе данных о приоритете, аварийности и количестве топлива сортирует список судов в порядке их вылета. Самолет с критически низким запасом топлива становится аварийным. Аварийные суда получают высокий приоритет.

Компонент определения доступности полос закрывает или вновь делает доступными полосы, основываясь на полученном из базы данных списком взлетных полос и прогнозе погоды, полученном после запроса к нейросети. При сильном ветре полосы, не совпадающие с направлением ветра, закрываются.

Компонент распределения нагрузки выводит финальный результат — порядок вылета самолетов, распределенных по конкретным доступным взлетным полосам.

Заключение

В ходе работы мною были изучены интеллектуальные системы и системы на основе знаний. Впервые был получен опыт создания и использования функциональной модели, IDEF0 нотации, Turtle Notation и SPARQL-запросов.

В результате был реализован макет интеллектуальной системы, который способен корректно обслуживать воздушное движение.

Приложение

Исходный код программы: <https://github.com/kseniadumpling/airport-intelligent-systems>