**МИНИСТЕРСТВО ОБРАЗОВАНИЯ РЕСПУБЛИКИ БЕЛАРУСЬ**

**Белорусский государственный Университет**

**ФАКУЛЬТЕТ ПРИКЛАДНОЙ МАТЕМАТИКИ И ИНФОРМАТИКИ**

**Кафедра многопроцессорных систем и сетей**

ЕФИМИК

Виталий Олегович

**СОЗДАНИЕ ЛОГИСТИЧЕСКОЙ СИСТЕМЫ МОРСКОГО ТРАНСПОРТА ДЛЯ КРУПНОГО БИЗНЕСА**

Отчёт о преддипломной практике

студента 5 курса 1 группы

|  |  |  |
| --- | --- | --- |
| “Допустить к защите”  с предварительной оценкой \_\_\_  Руководитель практики  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  “\_\_” \_\_\_\_\_\_\_\_\_\_\_\_\_\_ 2016 г. |  | Научный руководитель:  ассистент кафедры МСС  А.С. Гусейнова |
|  |  |  |

Минск, 2016

**РЕФЕРАТ**

Преддипломная практика, 30 с., 11 рис., 2 табл., 3 источника.

*Ключевые слова*: ТРАНСПОРТНЫЕ ПЕРЕВОЗКИ, МУЛЬТИПЛАТФОРМЕННОСТЬ, РЕПОРТИНГ, ЗАДАЧА ПРОГНОЗИРОВАНИЯ, МОДЕЛИ И МЕТОДЫ ПРОГНОЗИРОВАНИЯ, МЕТОД ПРОГНОЗИРОВАНИЯ SSA.

*Объект исследования* – концепции и методы мультиплатформенной разработки, применение метода SSA в задачах прогнозирования.

*Методы исследования* – обзор и изучение метода SSA, получение теоретический и практических знаний в гибкой мультиплатформенной разработке.

*Цель работы* – разработка транспортной логистической системы.

*Результатом* является реализованная транспортная логистическая система, изучение и освоение метода SSA в задачах прогнозирования.

*Областью применения* является транспортировка грузов и прогнозирование задержек.

ОГЛАВЛЕНИЕ

[ВВЕДЕНИЕ 4](#_Toc443589265)

[1 СИСТЕМА ГИС 5](#_Toc443589266)

[1.1 Разработка приложений 5](#_Toc443589267)

[1.2 Разработка компонент 6](#_Toc443589268)

[1.3 Разработка репортов 9](#_Toc443589269)

[2 МОДЕЛИ И МЕТОДЫ ПРОГНОЗИРОВАНИЯ 11](#_Toc443589270)

[2.1 Основные модели и методы прогнозирования 11](#_Toc443589271)

[2.2 Сингулярный спектральный анализ 12](#_Toc443589272)

[2.3 Прогноз с помощью метода SSA 16](#_Toc443589273)

[3 ПОСТАНОВКА ЗАДАЧИ И РЕАЛИЗАЦИЯ 18](#_Toc443589274)

[3.1 Модель данных 18](#_Toc443589275)

[3.2 Реализованная система 21](#_Toc443589276)

[ЗАКЛЮЧЕНИЕ 29](#_Toc443589277)

[Список использованных источников 30](#_Toc443589278)

## ВВЕДЕНИЕ

Транспортные перевозки – актуальная тема на сегодняшний день. Каждый день огромное количество всевозможных грузов необходимо доставить из одной точки в другую. Перевозка грузов востребована во многих отраслях: строительство, производство вычислительной техники, добыча полезных ископаемых, почта, пищевая промышленность и т.д. По всему миру постоянно идет производство товаров или, например, добыча ископаемых. Таким образом, необходимость доставить товар из точки “А”, в точку “Б” возникает постоянно. При этом зачастую объем данного товара настолько велик, что перевозки самолетом становятся невыгодными, например, межконтинентальные перевозки, а иногда – даже невозможными. Таким образом, оптимальными становятся перевозки путем морского транспорта. Т.к. объем перевозимого груза судов во много раз превышает тот, что может перевести иной транспорт. Однако при перевозках морским транспортом возникают и некоторые сложности. Например, такие как поломки транспорта. Но чаще всего на морской транспорт влияет погода. В океане часто разыгрываются шторма и бури, которые задерживают морской транспорт. Если какой-нибудь корабль приходит в порт с задержкой, и при этом контейнер с этого корабля должен быть перегружен на другой корабль, то данная задержка выбьет из графика и второй корабль тоже. Таким образом, появляется проблема предсказания задержек для морского транспорта. Ведь если заранее знать о том, что вероятно возникновение задержки в тот или иной период, то можно увеличить работу двигателей и избежать задержки прибытия грузов.

Использование анализа сингулярного спектра является хорошим и, на сегодняшний день, одним из самых популярных способов для решения задачи предсказания.

Анализ сингулярного спектра (Singular spectrum analysis) – метод анализа временных рядов, основанный на преобразовании одномерного временного ряда в многомерный ряд с последующим применением к полученному многомерному временному ряду метода главных компонент.

# СИСТЕМА ГИС

Система ГИС представляет собой мультиплатформенную систему для создания надежных клиент-серверных приложений. С помощью данной системы разрабатываются как клиентская часть, так и серверная часть. Информационная система ГИС архитектурно состоит из двух частей – это само приложение и компоненты, с которыми данное приложение общается.

### Разработка приложений

В данной информационной системе Клиентская часть, называемая Graphical User Interface (GUI), может представлять собой Web-страницу или же оконное приложение. Клиентская часть полностью реализуется на языке Java, разработка приложений на данной системе происходит в среде Eclipse, с помощью применения соответствующих плагинов. Для каждого GUI реализуется свой *контроллер*, который прослушивает java-компоненты данного приложения. Контроллер представляет собой Java класс, в котором описано множество методов для обработки стандартных Java событий java.awt.AWTEvent. Методы, реализованные в контроллере можно связать с любой java-компонентой. Возможны различные варианты связок методов контроллера с java-компонентами, т.е. один метод может прослушивать несколько java-компонент и одну java-компоненту может прослушивать несколько контроллеров. При возникновении события на java-компоненте, по порядку вызываются все методы, прослушивающие данную java-компоненту. Контроллер и GUI выполняются на стороне клиента. Система ГИС предоставляет множество java-компонент для реализации клиентского интерфейса, такие как кнопки, таблицы, лейблы, различные текстовые поля и редакторы, просмотрщик для PDF-документов, просмотрщик карты, поля для графиков, поля для просмотра времени и иные компоненты для управления и работы с данными. С помощью специальных плагинов в среде Eclipse возможен просмотр и редактирование графического интерфейса приложения. В редакторе графического интерфейса возможно устанавливать любое расположение предоставляемых java-компонент. Так же в редакторе производится установка всех свойств выбранных компонент. Например, таких как возможность редактирования полей, названия лейблов, возможность фокусировки и т.д. Кроме того система ГИС поддерживает создание логических условий. Логические условия чаще всего применяются для того, чтобы запретить или разрешить редактирование полей, сделать их фокусируемыми или запретить фокусировку или разрешить или запретить различные действия на пользовательском интерфейсе. Логические условия вычисляются динамически при возникновении различных событий на пользовательском интерфейсе, результат их вычисления зависит от содержимого полей данного приложения.

Контролер в свою очередь связан с *сервисом*. Данная связь происходит посредством сети. Сервис является связующим звеном между клиентской частью и сервером. Сервис так же реализован на языке Java и выполняется на отдельном сервере с помощью программного продукта IBM WS Application Server.

Данный продукт является промежуточным программным обеспечением, которое позволяет приложениям работать на разных платформах на основе веб-технологий. WS использует открытые стандарты XML[1].

С помощью сервиса производится отдельная часть реализации бизнес логики приложения. А именно производится проверка данных, введенных пользователем. Например, это может быть проверка допустимых комбинаций данных. Так же с помощью сервиса можно запустить другое клиентское приложение. При этом есть возможность автоматически заполнить в новом приложении различные поля, основываясь на данных из предыдущего приложения. Сервис представляет собой java-класс, в котором описаны различные методы. Методы сервиса удаленно вызывает контроллер.

Далее сервис вызывает серверную часть приложения с помощью специальных компонент называемых *бинами*. Бины представляют собой несколько java-классов и служат для связи между различными системами. Таким образом, возможно мультиплатформенное программирование. В системе ГИС клиентская часть полностью реализована на языке Java, а серверная на языке Ген, поддерживаемом компанией СА.

### Разработка компонент

Разработка сервера производится на языке Ген. Язык Ген является объектным и высоко структурированным. Т.е. в данном языке все является объектом и существует жесткая структура написания программ на данном языке. Структурированность данного языка помогает избегать синтаксических и большую часть логических ошибок. Для данного языка поддерживается специальная среда, которая помогает разработчикам в создании и управлении крупномасштабными мультиплатформенными приложениями. С помощью данного языка происходит описание архитектуры приложения и последующая генерация структурного скелета приложения и его связующих звеньев. В случае системы ГИС производится описание контроллера, сервиса и бинов. Среда разработки использует гибкие методологии для дизайна, развертывания и управления приложениями. Т.е. с помощью данного языка и данной системы возможна перегенерация всей архитектуры приложения в любой момент разработки и адаптация процесса разработки к новым требованиям.

При развертывании на сервере программы, написанной на языке Ген – происходит преобразование исходного кода языка Ген в низкоуровневый код на языке COBOL, после чего выполняется компиляция, линковка и запуск программы на сервере[2]. Для развертывания сервера используется мэинфреим под управлением операционной системы для серверов z/OS, которую разрабатывает и поддерживает компания IBM.

Язык Ген является процедурным. Таким образом, в языке Ген программа описывается в виде процедуры, из процедуры могут вызываться блоки кода и другие процедуры. В процессе выполнения процедуры происходит работа с объектами. Так же стоит отметить, что у каждой процедуры есть *импорты* и *экспорты*. Импорты – те объекты, которые подаются процедуре на вход. Экспорты – те объекты, которые процедура возвращает.

Архитектура всего приложения разрабатывается на языке Ген с помощью процедур следующим образом. Изначально разрабатывается процедура, в которой описываются события соответствующие методам сервиса. В данных событиях происходит различная установка системных данных и, при необходимости вызова сервера, описывается, какой сервер должен быть вызван при возникновении данного события. Далее происходит описание процедуры, соответствующей контроллеру. В данной процедуре, для описания методов контроллера, так же используются события. В каждом событии контроллера указывается то, какие события из сервиса необходимо вызвать в процессе обработки. Так же устанавливается различная системная информация. Если возникает такая необходимость – можно из контроллера произвести вызов сервера, миновав сервис. В экспортах данной процедуры происходит определения объектов, которые будут пересылаться между клиентской частью и сервером, а в импортах – описание объектов, которыми можно инициализировать данное приложение для оперативного открытия через сервис (вызов данного приложения из других приложений).

Система ГИС предоставляет специальный плагин, который производит чтение и анализ описанных выше процедур, написанных на языке Ген. Рассматриваемый плагин предназначен для генерации каркаса приложения, т.е. для создания структуры всех java-классов данного приложения. Например, таких как контроллер или сервис. Так же данный плагин предназначен для полной или частичной перегенерации архитектуры приложения. Данная способность позволяет поддерживать возможность разработки с применением гибких методологий, в которых требования к дизайну и архитектуре конечного продукта постоянно меняются.

Сам сервер описывается на языке Ген в виде процедуры. Из данной процедуры может происходить вызов отдельных блоков кода и других процедур, называемых *операциями* или *компонентами*. Блоки кода обычно используются для обобщения выполняемых действий, например, таких как управление данными (создание, модификация, удаление) или считывание данных. Операции предназначены для программной реализации работы с данными, т.е. непосредственного обращения к серверам с базами данных и выполнения SQL-запросов в базе данных DB2. Так же операции могут выполнять преобразование данных и возвращение определенных значений, например, текущее время или дата. При работе с базами данных используются специальные объекты-сущности, которые соответствуют таблицам базы данных. Архитектурно операции состоят из следующих блоков:

1. интерфейс;
2. маппер;
3. блоки копирования;
4. имплементация.

При вызове какой-либо компоненты используется интерфейс. Интерфейс разрабатывается следующим образом. Изначально в интерфейсе определяется, какие объекты принимает данная компонента и какие возвращает. А программная реализация состоит только из вызова маппера. После этого происходит компиляция и линковка интерфейса. Далее из интерфейса удаляется маппер, и новая версия интерфейса закачивается на сервер, но не происходит компиляции и линковки. Если не удалить маппер, то при импортировании компоненты в другие пакеты пришлось мы импортировать все блоки кода. А таким образом, если маппер удален, необходимо импортировать только пустой интерфейс. При этом остальные блоки кода содержатся только в исходном пакете. Однако при вызове интерфейса будет выполняться его последняя откомпилированная версия, в которой маппер содержится.

Маппер выполняет поочередный вызов блоков копирования и блоков имплементации. Блоки копирования выполняют преобразование обычных объектов в объекты-сущности, с помощью которых происходит обращение и работа с базой данных, и наоборот. А блок имплементации выполняет непосредственное обращение и работу с базой данных[].

### Разработка репортов

Система ГИС так же предоставляет возможность оперативного извлечения информации из базы данных и представления этой информации в удобном виде для оценки и наглядного восприятия. Для это служат отдельные приложения, называемые *репортами*. Разработка данных приложений так же производится с помощью среды Eclipse с использованием специальных плагинов.

Разработка репортов производится с помощью специального скриптового языка баттерфляй и SQL-запросов. Каждый репорт содержит основной скрипт, который выполняется при запуске репорта. Так же репорт может содержать неограниченное количество SQL-запросов. В процессе выполнения основного скрипта происходит вызов SQL-запросов в любой последовательности и постепенное заполнение выходных данных. Выходные данные основного скрипта обычно представляются в виде таблицы. Т.е. происходит постепенное (построчное) заполнение выходной таблицы. В процессе заполнения каждой строки у разработчика есть возможность производить различные операции с данными, выполнять их преобразование или же выполнять другие запросы аналогичным способом. Таким образом, у разработчика появляется гибкий инструмент для выборки данных из таблицы, при использовании которого появляется возможность эффективно использовать SQL-подзапросы.

Перед выполнением запросов с помощью скриптового языка есть возможность произвести проверку введенных данных на разрешенные значения или ошибки.

При проверке и обработке данных разработчику предоставляются удобные библиотеки для модификации и обработки данных. Система ГИС располагает обширными библиотеками для работы с:

1. числами;
2. датами и временем;
3. строками;
4. SQL-запросами;
5. групповой обработки данных.

Данные для репорта вводятся с помощью специального диалогового окна, которое отображается при запуске скрипта.

После выполнения основного кода результат записывается в XML-файл. При необходимости произвести специфическую обработку данных, например нетривиальную сортировку, полученный XML-файл обрабатывается с помощью еще одного преобразующего скрипта, написанного на рассматриваемом языке. После всех обработок данные выдаются в заранее заданном виде. Для отображения данных могут быть созданы шаблоны следующих форматов XML, EXCEL (XLS, XLSX), PDF. Один репорт может содержать несколько форматов вывода, и каждый формат может содержать свой преобразующий скрипт. При этом при создании шаблонов, возможно использование всех функций и преимуществ выбранного формата. Например, для EXCEL документов с помощью макросов можно произвести построение “на лету” различных графиков, диаграмм, PivotTables и т.д. с выбранными данными.

# МОДЕЛИ И МЕТОДЫ ПРОГНОЗИРОВАНИЯ

### Основные модели и методы прогнозирования

Модель прогнозирования – некоторое представление объекта прогнозирования, которое описывает исследуемый процесс. Исследование модели прогнозирования позволяет получить информацию о возможных состояниях объекта в будущем.

Метод прогнозирования – способ исследования объекта прогнозирования, направленный на разработку прогноза. Методы прогнозирования состоят из последовательности действий, которые необходимо совершить для получения той или иной модели прогнозирования.

Методы прогнозирования делятся на две группы: интуитивные и формализованные. В интуитивных методах используются суждения и оценки экспертов. Сферы применения интуитивных методов: маркетинг, экономика, политика и т.п. Таким образом, прогнозирование интуитивными методами используется либо в сферах, которые очень сложно описать математически, либо в очень простых сферах, когда нет необходимости в описании с помощью математической модели. В обратном случае применяются формализованные методы. При применении формализованных методов, строятся модели прогнозирования. Т.е. происходит определение некоторой математической зависимости, с помощью которой возможно вычисление будущего значения исследуемого процесса.

Формализованные методы включают в себя две группы моделей: модели предметной области и модели временных рядов.

Модели предметной области относятся к конкретной сфере применения, в которой существуют специфические уравнения или законы, которые необходимо учитывать для рассмотрения и исследования данной области. Примером можно привести прогноз погоды, в нем используются уравнения термодинамики и динамики жидкостей.

Модели временных рядов – это математические модели прогнозирования, с помощью которых возможно нахождения зависимости будущего значения исследуемого процесса от прошлых значений. На основе этой зависимости производится вычисление прогноза. Модели временных рядов можно считать универсальными, поскольку их внешний вид не зависит от природы происхождения временного ряда и остается неизменным. Зависимость между будущим значением и прошлыми значениями выражается в виде аппроксимирующей функции.

В зависимости от области определения параметров аппроксимирующей функции выделяют две основные группы моделей: глобальные и локальные.

В глобальных моделях параметры аппроксимирующей функции идентифицируются посредством использования всех известных значений ряда. Основная цель глобальных моделей – получение глобальных характеристик системы. К ним относятся: авторегрессионные модели, сингулярный спектральный анализ.

Локальные модели основаны на принципе локальной аппроксимации (LA). Применяются в целях прогнозирования динамики ряда.

Стоит отметить, что на сегодняшний день для прогнозирования временных рядов имеется огромное количество моделей и методов для построения прогноза, например, таких как SVM модели, модели на основе генетических алгоритмов и многие другие.

### Сингулярный спектральный анализ

Метод сингулярного спектрального анализа (SSA) используется для определения основных составляющих временного ряда и подавления шума. Метод SSA позволяет:

1. различать составляющие временного ряда, полученные из последовательности значений какой-либо величины;
2. находить заранее неизвестные периодичности ряда;
3. сглаживать исходные данные на основе отобранных составляющих;
4. наилучшим образом выделять компоненту с заранее известным периодом;
5. предсказывать дальнейшее поведение наблюдаемой зависимости.

В основе SSA лежит построение множества векторов задержек:

|  |  |
| --- | --- |
|  | (1) |

Метод задержек устанавливает переход от исходного одномерного (скалярного) временного ряда к многомерному (векторному) представлению. Каждый многомерный вектор образуется из некоторого числа p следующих друг за другом значений исходного временного ряда:

|  |  |
| --- | --- |
|  | (2) |

Здесь каждая квадратная скобка — вектор в p–мерном пространстве задержек; последовательность таких векторов задает матрицу задержек , где *N* — число элементов исходного ряда.

Особенностью SSA является обработка матрицы X по алгоритму, похожему на метод главных компонент[3]. Суть метода главных компонент состоит в снижении размерности исходного пространства факторов (задержек) с помощью ортогонального линейного преобразования. Полученные таким образом новые переменные и называют главными компонентами. Применение этого метода позволяет сгладить исходный ряд, снизить уровень случайных возмущений, повысить отношение сигнал/шум.

Способы прогнозирования на базе SSA следующие: (а) “Гусеница” и (б) метод авторегрессии, который применяется по отдельности к каждой из выбранных компонент разложения. Ниже представлены основные этапы применения SSA к конкретному ряду .

Изначально происходит преобразование одномерного ряда в многомерный. Для такого преобразования необходимо взять некоторое число задержек .

|  |  |
| --- | --- |
|  | (3) |

|  |  |
| --- | --- |
|  | (4) |

Построенная матрица X является прямоугольной, но в предельном случае, т.е. при τ = (N + 1)/2 и нечетном N, она вырождается в квадратную.

Далее происходит построение для матрицы X соответствующей ковариационной матрицы:

|  |  |
| --- | --- |
|  | (5) |

После чего происходит определение собственных значений и собственных векторов матрицы *С*. Для этого ее необходимо разложить по собственным векторам:

|  |  |
| --- | --- |
|  | (6) |

Где введены следующие матрицы: диагональная матрица собственных чисел Λ и ортогональная матрица собственных векторов матрицы C.

|  |  |
| --- | --- |
|  | (7) |

|  |  |
| --- | --- |
|  | (8) |

Таким образом, будут выполняться следующие равенства:

|  |  |
| --- | --- |
|  | (9) |

|  |  |
| --- | --- |
|  | (10) |

последнее равенство справедливо только в случае предварительно нормированных строк матрицы X.

Далее происходит переход к главным компонентам. Матрицу собственных векторов V обычно представляют как матрицу перехода к главным компонентам исходного ряда.

|  |  |
| --- | --- |
|  | (11) |

Где величины Yi , i = 1, 2, . . . , τ , — строки длины n. При этом собственные значения можно рассматривать как вклад главных компонент в общее информационное содержание временного ряда . Тогда по полученным главным компонентам можно полностью восстановить исходную матрицу

|  |  |
| --- | --- |
|  | (12) |

а по ней в свою очередь можно восстановить временной ряд .

Преобразования , i = 1, 2, . . . , τ , , *l* = 1, 2, . . . , *n*, являются *линейными фильтрами*.

Каждый *i*-й собственный вектор включает в себя *τ* компонент, т.е. . Построим зависимость значений компонент , k = 1, 2, . . . , *τ* , от их номера: . Тогда, используя свойство ортогональности собственных векторов, дальнейший анализ последовательности можно провести, изучая диаграммы, построенные по аналогии с *фигурами Лиссажу*. Именно по осям попарно откладываются компоненты , . Если построенные диаграммы оказываются близкими по форме к окружности, то функции , будут подобны периодическим с близкими амплитудами и со сдвигом фазы около четверти периода.

Таким образом, для некоторых пар собственных векторов Vi , Vj можно вычислить величину, имеющую смысл периода. Следовательно, графический анализ дает представление о частотах компонент, формирующих исходный временной ряд .

Далее возможно произвести SSA-сглаживание. Предположим, что из τ компонент для дальнейшего анализа оставлены лишь первые *r*. Тогда для восстановления исходной матрицы X используются первые *r* собственных векторов . В таком случае

|  |  |
| --- | --- |
|  | (13) |

где – восстановленная матрица, имеющая *n* столбцов и τ строк. Теперь исходный временной ряд, воссозданный из этой матрицы, определится как

|  |  |
| --- | --- |
|  | (14) |

Данный способ получения последовательности называется SSA-сглаживанием исходного временного ряда по первым *r* компонентам из τ .

### Прогноз с помощью метода SSA

На этом этапе строится ряд . При этом прогноз на p точек вперед осуществляется как применение p раз операции прогноза на одну точку.

Базовая идея нахождения значения состоит в следующем. Пусть имеется набор значений , , … , . Теперь построим выборку в виде матрицы X. В качестве базиса поверхности, содержащей эту выборку, можно взять отобранные ранее собственные векторы , , . . . , матрицы C.

Запишем параметрическое уравнение этой поверхности как , где каждому значению вектора *S(P)*, представляющего собой столбец высотой τ , соответствует набор из *r* параметров . Тогда *k*-му (*k* = 1, 2, . . . , *n*) столбцу исходной матрицы X соответствует свой набор значений параметров = ( , , . . . , ) и, следовательно,

|  |  |
| --- | --- |
|  | (14) |

Для предсказания значения необходимо построить (*n + 1*)-й столбец матрицы X, которому соответствует некоторое значение параметров . Этот набор параметров можно найти из соотношения , исходя лишь из данных . При этом прогнозируемый столбец будет записан следующим образом: .

Введем следующие обозначения:

|  |  |
| --- | --- |
|  | (14) |

|  |  |
| --- | --- |
|  | (14) |
|  |  |

Параметры можно определить из системы уравнений относительно . Таким образом, окончательное выражение для прогнозируемого значения будет выглядеть как

|  |  |
| --- | --- |
|  | (14) |

Для предсказания следующих значений в простейшем случае требуется лишь изменить соответствующим образом матрицу *Q* и вновь умножить ее на величину / ( ). Однако дополнительно при этом можно для каждой следующей точки повторять частично или полностью весь алгоритм метода SSA. В этом случае будут также меняться матрицы и .

# ПОСТАНОВКА ЗАДАЧИ И РЕАЛИЗАЦИЯ

В качестве проекта было решено реализовать транспортную логистическую систему для морского транспорта, которая бы позволяла отображать положение кораблей, отображать перегрузки грузов, а так же отображать и оценивать задержки в пути следования кораблей.

Оценку задержек было решено производить при помощи алгоритма SSA, на сегодняшний день он является одним из самых популярных алгоритмов в прогнозировании.

Для реализации транспортной логистической системы была выбрана система ГИС. Данная система предоставляет удобный и гибкий функционал для создания приложений. Клиентская часть реализована на языке Java, сервер реализован на языке Ген.

### Модель данных

Для реализации данной логистической системы было принято участие в создании модели данных. Для хранения данных о кораблях было принято решение создать модель данных следующим образом. В отдельной таблице хранится информация о корабле, в отдельной таблице хранится информация о местоположении кораблей. Связь реализуется один ко многим, таким образом, предоставляется возможность хранить историю следования корабля. Данная модель представлена на рисунке 3.1.

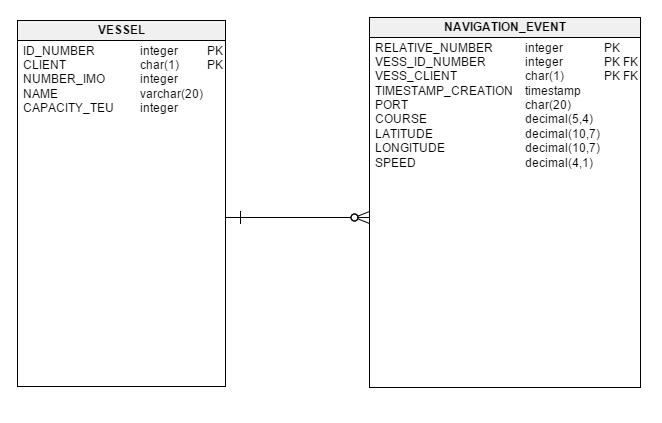


Рисунок . – модель данных “корабль-событие”

Для хранения данных о портах был выбран аналогичный подход. В отдельной таблице хранятся данные о терминале, в отдельной таблице хранятся данные о порте, в отдельной таблице хранятся данные о расположении. Таблицы порт и терминал связаны отношение один ко многим, т.к. в одном порту может быть несколько терминалов. Таблицы порт и местонахождение связаны один к одному. Данную модель данных можно увидеть на рисунке 3.2.

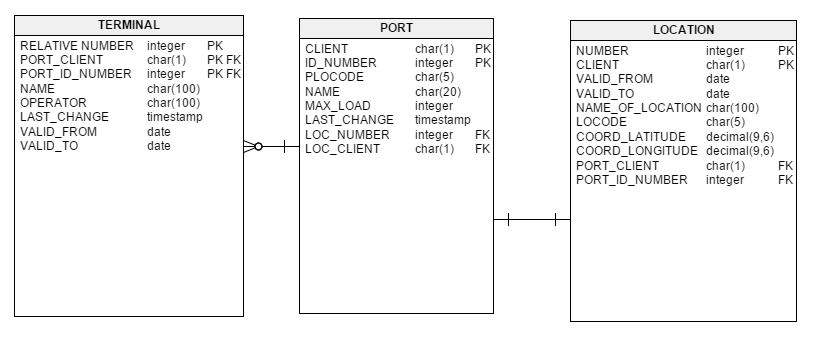


Рисунок . – модель данных “терминал-порт-местонахождение”

В данной работе так же отражена проблема задержек кораблей. В судоходстве необходимо оценивать задержки для предотвращения ошибок в расписании и запаздывании доставок грузов. Система ГИС поддерживает программы типа репорт, которые служат для оперативного предоставления информации. При проектировании репортов часто возникает необходимость учитывать новые специфические данные, для которых не предусмотрена архитектура базы данных. Например, некоторые вычисленные значения, или цвета для отображения уровней опасности. Тогда для ввода данных пришлось бы перепроектировать базу данных, но такой подход может оказаться неэффективным и трудозатратным. Поэтому, для вывода данных о задержках в отчет, было принято решение в отдельной таблице хранить список уникальных задержек, а данные об этих задержках хранить в отдельной таблице, при этом связь между таблицами установить один ко многим. Т.е. в таблице с данными будет два столбца характеризующих задержку, а именно: название атрибута и значение атрибута. А таблица с уникальными значениями будет содержать идентификационный номер задержки. Таким образом, возможно добавление все новой и новой полезной информации в базу данных относительно конкретной задержки.

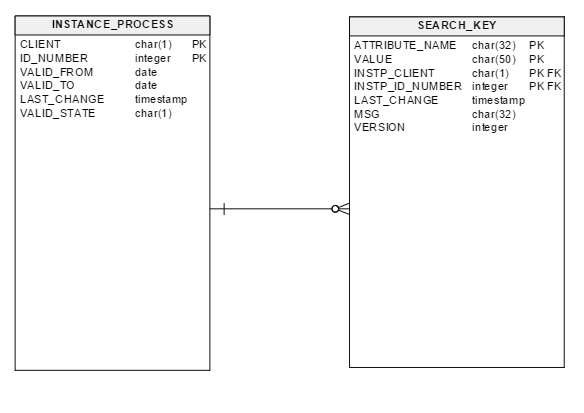


Рисунок . – модель данных “процесс-значение”

При перевозке груза из одного места в другое, обычно используется более одного корабля. Каждый корабль плывет по заранее заданному маршруту, при этом, приплывая в порт, происходит выгрузка некоторых привезенных грузов и загрузка новых грузов. Если же какой-нибудь из кораблей задерживается, то есть вероятность, что грузы, которые с него необходимо выгрузить, опоздают на загрузку на новый корабль. Таким образом, необходимо учитывать количество грузов, которые планируется загрузить на данный корабль и выгрузить с данного корабля. Каждому кораблю назначается свой маршрут. Маршруты, корабли и количество разгруженных и загруженных грузов хранится в разных таблицах. В таблице с количеством грузов, для большей эффективности, хранится порт, исходный маршрут и маршрут, на который грузы перегружаются.

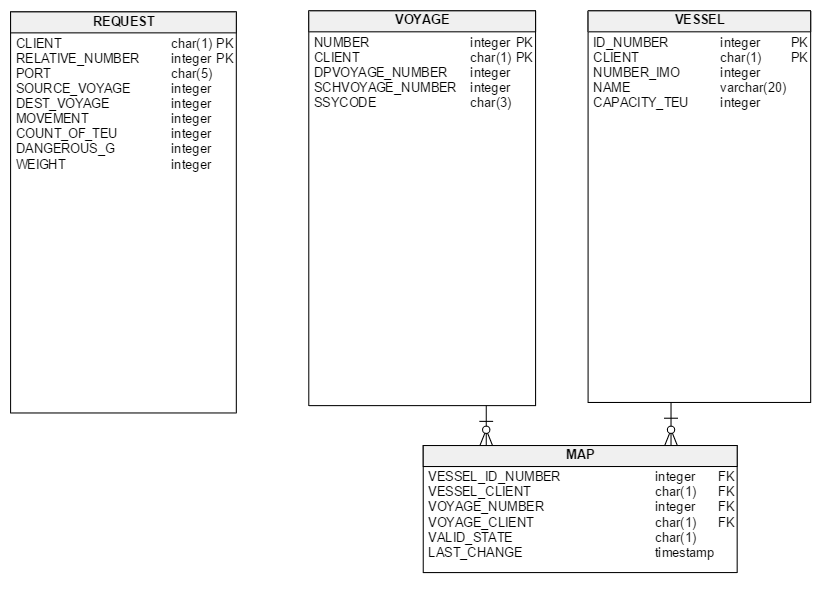


Рисунок . – модель данных “грузы-маршруты-корабли”

Модель данных реализована в базе данных DB2.

### Реализованная система

Для отображения положения кораблей, портов и терминалов было реализовано приложение, которое, с помощью доступного API, отображает карту, извлекает из базы данных информацию о местоположении интересующих кораблей, портов или терминалов и обозначает данное положение на карте соответствующими знаками. Карту предоставляет система ГИС. В построенной модели данных сохраняются широта и долгота положения всех перечисленных объектов (терминалы, порты и корабли). При этом так же хранится информация о каждом из объектов. После выборки этой информации из базы данных формируется XML-файл, и, с помощью соответствующего API, данный XML файл устанавливается в карту, предоставляемую системой ГИС. После всех действий система ГИС возвращает изображение карты, при этом в заданных положениях широты и долготы отображаются соответствующие объекты.

В реализованном приложении предоставляется возможность задания списка интересующих кораблей. Возможно задание идентификационного номера или имени корабля. Так же можно произвести поиск порта или терминала на карте. Поиск производится по введенным кодам портов, которые являются уникальными и хранятся в базе данных. Порты наносятся на карту аналогичным методом.

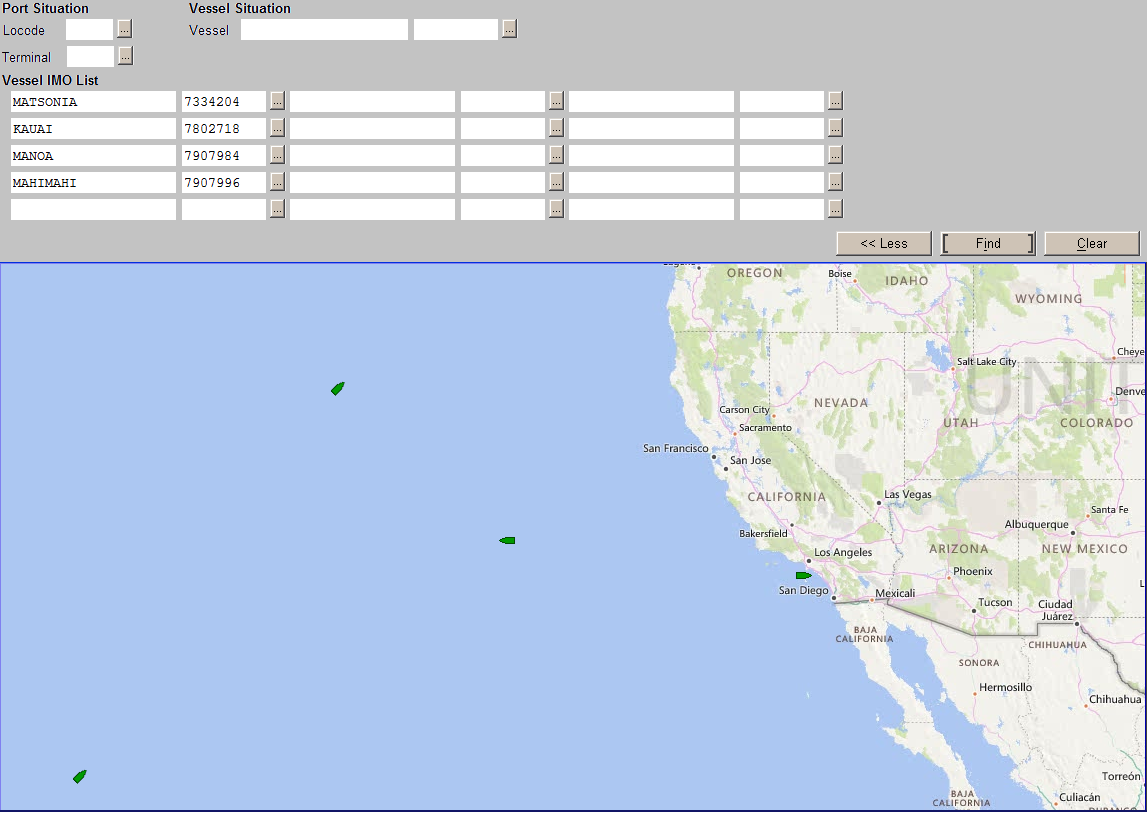


Рисунок . – приложение “карта”

Так же было реализовано приложение, которое отображает связь между кораблями. В данном приложении отображается запланированное количество грузов, которые необходимо перегрузить с одного корабля на другой. Соответствующее приложение помогает оценить, насколько корабли связаны друг с другом. Эти данные являются важными при оценке того, насколько велико влияние задержки корабля. Например, пусть данный корабль (корабль А) везет большое количество грузов, которые необходимо перегрузить на другой корабль (корабль Б). Пусть время t – время прибытия, корабля А в порт, а время s – время отправления корабля Б из этого порта. Таким образом чем ближе время t ко времени s – тем меньше остается времени на перегрузку и тем больше вероятность не успеть перегрузить все контейнеры с корабля А на корабль Б. Если же корабль А задержится и время t станет больше, чем время отплытия корабля Б. То контейнеры остаются в порту и ожидают следующего корабля, движущегося по маршруту, аналогичному тому, на который контейнер опоздал. Таким образом, информация о связности кораблей по контейнерам может быть ключевой при решении увеличить работу двигателей и прибыть в порт, минуя задержки. В реализованном приложении есть возможность ввести интересующий корабль, время его прибытия в порт и соответствующий порт. Или же интересующий идентификатор маршрута корабля и идентификатор порта в этом маршруте. После чего приложение отображает количество грузов, которые перегружаются с корабля, идущего по заданному маршруту на другие корабли и количество грузов которые загружаются на данный корабль и далее движутся по заданному маршруту.

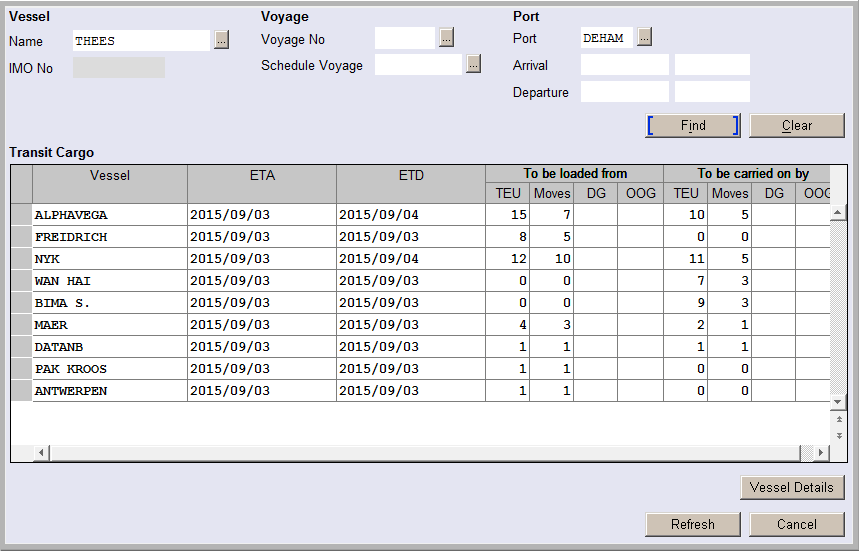


Рисунок .6 – приложение “связь между кораблями”

В реализованной системе находит отражение информация о задержках кораблей. Данная информация выводится с помощью программы типа репорт, реализованной с помощью средств и возможностей системы ГИС. При выполнении программы информация извлекается из разработанной модели данных. При запуске разработанной программы появляется диалоговое окно, в котором необходимо ввести данные для фильтрации информации.

Предоставляемые фильтры:

1. исходный регион – регион отправления корабля;
2. целевой регион – регион прибытия корабля;
3. направление движения;
4. нижняя граница времени отправления;
5. верхняя граница времени отправления;
6. нижняя граница времени прибытия;
7. верхняя граница времени прибытия.

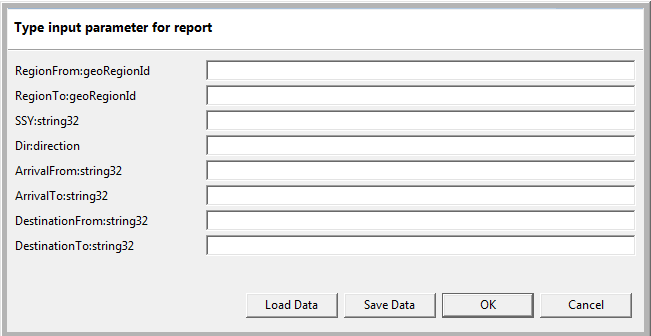


Рисунок .7 – приложение “репорт”, окно ввода параметров

После выполнения работы программы генерируется EXCEL документ, где отображена информация о задержках кораблей. Так же данный документ отражает информацию, которая позволяет произвести оценку для задержек. В документе указывается корабль, у которого возникла задержка и на какой части следования судна данная задержка возникла. Так же указывается величина этой задержки и когда именно произошла данная задержка. Таким образом, отображается информация, которую можно использовать для анализа и выявления проблемных мест в маршрутах.

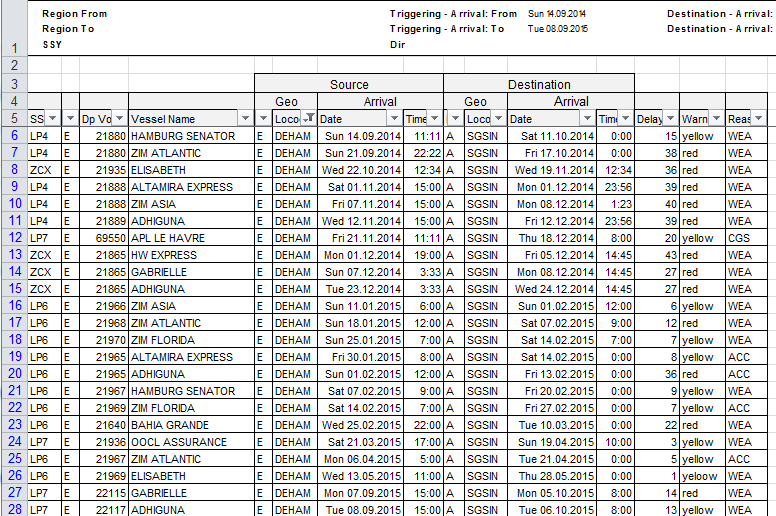


Рисунок .8 – результат работы приложения “репорт”

Задержки возникают очень часто, как показывает практика. Кроме того, задержки могут оказывать существенное влияние на работу всей системы. Поэтому, для более эффективного использования кораблей, необходимо производить анализ предыдущих значений задержек. Например, часто задержки возникают из-за погодных условий, которые подвержены периодичности. Алгоритм SSA на сегодняшний день является популярным алгоритмом для прогноза значений и применяется во множестве сфер. В данной системе был реализован алгоритм SSA для прогнозирования задержек.

Для прогнозирования были доступны данные о задержках для кораблей, следующих из Гамбурга в Сингапур. По расписанию, отправление по данному маршруту происходило 4 раза в месяц. Результат работы алгоритма представлен на рисунке 3.9.

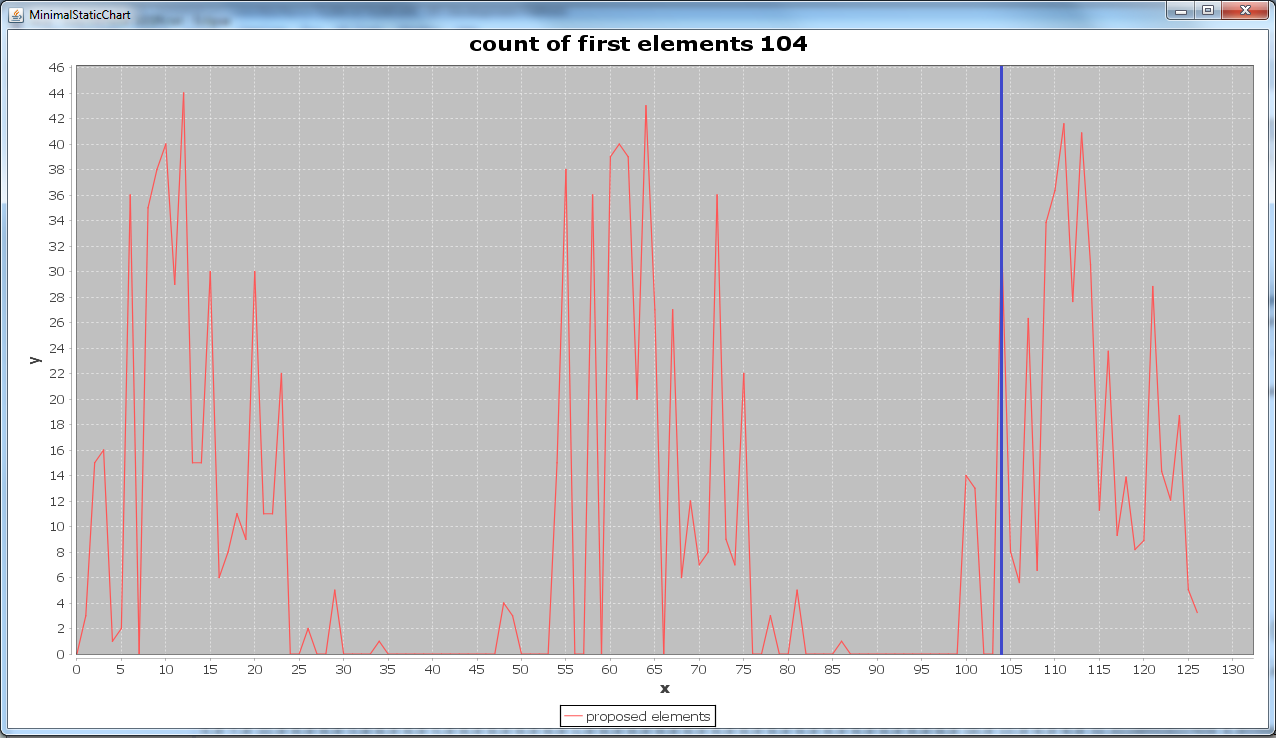


Рисунок .9 – прогноз задержек на зимний период

Таким образом, данная программа спроектировала будущие задержки на зимний период. Т.е. такие задержки ожидаются, если компания будет придерживаться такой же политики доставки грузов. А именно произойдет последовательность задержек, представленная в таблице 1. Из построенного графика и таблицы видно, что в зимний период ожидаются гораздо большие задержки, чем в летний период. Так же, на основании предыдущих задержек можно выявить в какие месяцы задержки будут наиболее сильными. Например, для данной части маршрута, наибольшие задержки случаются в ноябре и декабре. В январе же, задержки ожидаются в два раза меньше, чем в предыдущие два месяца. Таким образом, если связь между кораблями по грузам высокая, то необходимо принять решение об увеличении работы двигателей в эти месяцы.

Таблица 1 – спроектированная последовательность

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Октябрь | | | | ноябрь | | | | декабрь | | | | январь | | | | февраль | | | | ... | |
| 8 | 5 | 26 | 6 | 33 | 36 | 41 | 27 | 40 | 30 | 11 | 23 | 9 | 13 | 8 | 8 | 28 | 14 | 12 | 18 | 5 | 3 |

Так же производилось сравнение уже известных задержек с тем, что выдает алгоритм. Данное сравнение представлено в таблице 2 и на рисунке 3.10. При оценке данной таблицы можно судить о том, что в летние месяцы ожидаются небольшие задержки в границах до 4-х часов, но уже в сентябре и октябре величины задержек станут значительно увеличиваться.

Таблица 2 – спроектированная последовательность

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | Март | | | | Апрель | | | | Май | | | | Июнь | | | |
| Прогноз | 0 | 0 | 1 | 4 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| Реальные значения | 0 | 0 | 3 | 0 | 0 | 5 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 |
|  | Июль | | | | Август | | | | Сентябрь | | | | Октябрь | | | |
| Прогноз | 0 | 0 | 0 | 0 | 0 | 0 | 4 | 2 | 0 | 0 | 3 | 21 | 31 | 5 | ... | |
| Реальные значения | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 14 | 13 | 0 | 0 | 15 | 29 | ... | |

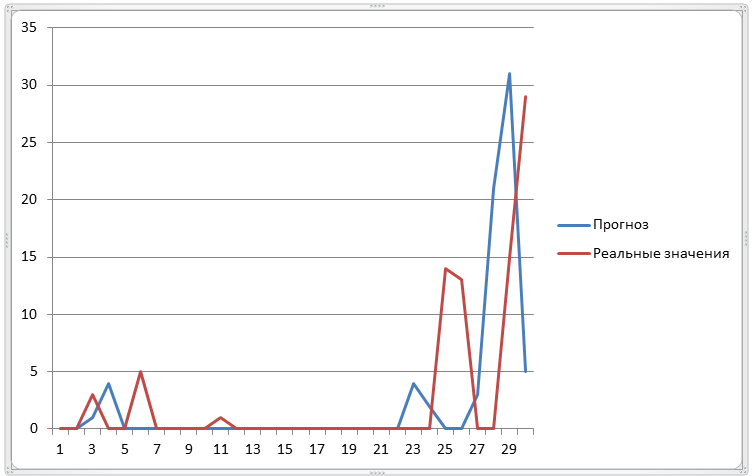


Рисунок .10 – сравнение прогноза с реальными значениями

Для тестирования реализованного алгоритма так же была использована база чисел Вольфа. Числа Вольфа – числа, характеризующие солнечную активность и количество пятен на солнце. Числа Вольфа измеряются каждый год и хранятся в общедоступной базе. Выявлено, что числа Вольфа носят периодический характер, следовательно, на них можно протестировать прогнозирующие алгоритмы. Эмпирическим путем установлено, что период чисел Вольфа равен примерно 11 годам. Результат тестирования алгоритма на числах Вольфа представлен на рисунке 3.11. Как видно из графика – первый спрогнозированный период достаточно точен, однако далее уже происходит расхождение.

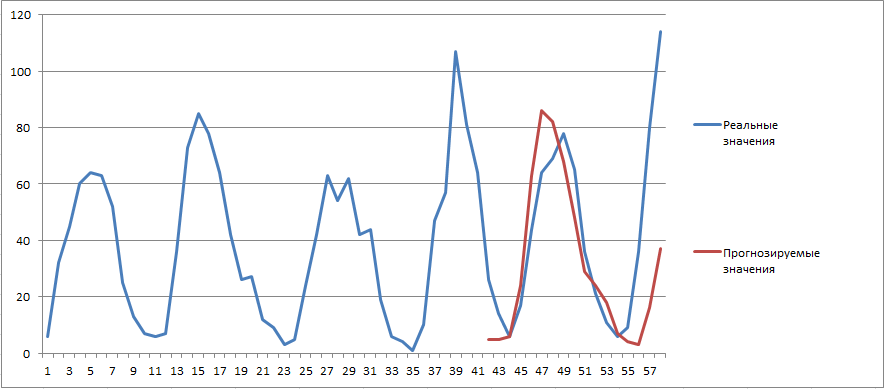


Рисунок .11 – сравнение прогноза с реальными значениями

## ЗАКЛЮЧЕНИЕ

В данной работе:

* изучены гибкие методы разработки приложений в мульти-платформенной системе ГИС;
* рассмотрен метод прогнозирования SSA;
* разработана модель данных для транспортной логистической системы в базе данных DB2;
* реализовано приложение для отображения кораблей, портов и терминалов на карте;
* реализовано приложение для отображения связи кораблей по грузам;
* реализовано приложение репорт для оперативной оценки задержек кораблей;
* произведена реализация алгоритма SSA;
* выполнено тестирование алгоритма SSA для задержек кораблей и для чисел Вольфа;

## Список использованных источников

1. Фрэнсис, Т. АЙ БИ ЭМ профессиональный WS сервер / Т. Фрэнсис. – Джон Войли и Сонс, 2004. - 792 с.
2. Райнс, К. Разработка и развертывание Java приложений/ К. Райнс. – АЙ БИ ЭМ Рэдбукс, 2013 – 406 с.
3. Лоскутов, А. Ю. Анализ временных рядов / А. Ю. Лоскутов. – Физфак МГУ, 2001. – 106 с.