

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ
УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ «РОССИЙСКИЙ ГОСУДАРСТВЕН-
НЫЙ ГИДРОМЕТЕОРОЛОГИЧЕСКИЙ УНИВЕРСИТЕТ»

На правах рукописи



Яготинцева Наталья Владимировна

**МЕТОДИЧЕСКОЕ ОБОСНОВАНИЕ ГЕОИНФОРМАЦИОННОЙ
СИСТЕМЫ ПОДДЕРЖКИ ПРИНЯТИЯ РЕШЕНИЯ ПРИ
УПРАВЛЕНИИ МОРСКИМ ДИНАМИЧЕСКИМ ОБЪЕКТОМ**

Специальность 25.00.35 - Геоинформатика

Диссертация на соискание ученой степени

кандидата технических наук

Научный руководитель:

д.т.н., доцент Т.М. Татарникова

Научный консультант:

к.в.н., доцент А.Г. Соколов

Санкт-Петербург – 2017

Оглавление

1. Анализ элементов геоинформационной системы поддержки принятия решения при управлении морским динамическим объектом	17
1.1 Корабль как морской динамический объект.....	17
1.2 Характеристика компонентов ГИС корабля.....	21
1.3 Информационное обеспечение ГИС корабля.....	23
1.4 Программное обеспечение ГИС корабля.....	30
1.5 Аппаратное обеспечение ГИС корабля.....	32
1.6 Структурно-функциональная модель ГИС корабля	33
1.7 Постановка задачи исследования.....	38
ВЫВОДЫ ПО ПЕРВОЙ ГЛАВЕ	41
2. Система моделей оценки передачи данных в гис корабля	42
2.1 Оценка времени установления соединения.....	42
2.1.1 Математическая модель установления соединения в ГИС корабля.	43
2.1.2 Эксперимент на модели.....	48
2.2 Оценка времени передачи данных	54
2.2.1 Математическое обеспечение расчета времени передачи данных ...	55
2.2.3 Эксперимент по оценке времени передачи данных	60
2.3 Оценка времени решения задачи.....	61
2.3.1 Математическое обеспечение расчета времени решения задачи	61
2.2.2 Эксперимент на модели.....	63
ВЫВОДЫ ПО ВТОРОЙ ГЛАВЕ	65
3 Методика проектирования структурно-функциональной модели гис корабля	66
3.1 Определение исходных данных для построения структуры системы.	67
3.2 Определение временных характеристик	71
3.3 Определение полного списочного состава.....	73
3.4 Шаги методики проектирования структурно-функциональной модели ГИС корабля	78
ВЫВОДЫ ПО ТРЕТЬЕЙ ГЛАВЕ.....	80

4 Экспертная система принятия решения по выбору структуры гис корабля .	81
4.1 Общая характеристика экспертной системы.....	86
4.2 Реализация экспертной системы	91
ВЫВОДЫ ПО ЧЕТВЕРТОЙ ГЛАВЕ	101
ЗАКЛЮЧЕНИЕ	102
ЛИТЕРАТУРА	104
ПРИЛОЖЕНИЕ	112

Введение

Внедрение на морские суда в 90-х годах прошлого века вычислительной техники и информационных технологий считается началом автоматизации судовождения. Последующее применение возможностей вычислительной техники в задачах управления морским судном или кораблем привело к появлению навигационных информационных систем, что позволило повысить оперативность и точность решения традиционных задач судовождения.

В XXI век мировое морское сообщество вступило с двумя принципиально новыми технологиями в области судовождения: глобальные спутниковые радионавигационные системы Глонасс, GPS и географические информационные системы (ГИС). [27,31,34]

Первая технология связана с получением в любой точке земной поверхности высокоточных координат движущегося объекта, вектора его абсолютной скорости, а также точного времени. Это открыло новые пути для организации более высокого уровня безопасности мореплавания на основе систем сопровождения, диспетчеризации, систем управления движением судов. Например, возможность непрерывного определения координат корабля открыла новые направления в автоматизации судовождения, такие как автоматическое нанесение на электронные планшеты и карты текущего места объекта, его траектории и автоматическое управление по заданному маршруту. Непрерывная трансляция в эфир судовых координат коренным образом изменила подход к решению задачи предупреждения столкновений судов. Знание штурманом скорости и направления абсолютного перемещения своего судна позволило по-другому подходить к ведению счисления пути. [62,64,65,77]

Технология ГИС сформировалась вместе с появлением быстродействующих ЭВМ и систем хранения данных. Географические информационные системы представляют собой базу данных с привязкой к географическим координатам. К таким данным поначалу относилась топографическая, гидрографическая и гидрологическая информация, гидрометеорологические особенно-

сти, видеоинформация, дающая изображения мысов, приметных объектов и т.д., информация о радионавигационных средствах, огнях, знаках, подводных коммуникациях и т.д. С помощью появившихся программных средств эта информация дополнялась, корректировалась, передавалась в другие базы данных, обрабатывалась и представлялась в виде, удобном для того или иного применения. В результате такой практической работы в настоящее время мы имеем ГИС с полным набором функций, примером которых могут служить современные ECDIS (Electronic Chart and Display Systems).[11,13,20,46]

Современная концепция обеспечения безопасности управления динамическими объектами предполагает необходимости дополнения координатных характеристик динамических объектов соответствующими атрибутивными (некоординатными) характеристиками, включающими технические параметры систем, управляющие сигналы, динамику реализации сигналов управления и др. Такой подход широко применяется для различных объектов – авиационных, автомобильных, морских и др. Значимость учета атрибутивных характеристик возрастает с ростом сложности объекта и системы принятия решений. Морские динамические объекты среди перечисленных представляются наиболее сложными и роль человеческого фактора для управления ими является наиболее значимым.

Интегрирование навигационных информационных систем с системами управления морскими динамическими объектами открывает еще более широкие возможности по обеспечению безопасности плавания и автоматизации процесса судовождения с учетом атрибутивных (некоординатных) характеристик корабля. Основаниями для разработки таких систем послужило:

во-первых, расширение понятия «морской динамический объект» с учетом необходимости включения в процесс принятия решений атрибутивных (некоординатных) характеристик;

во-вторых, принятие международных стандартов в области электронной картографии и высокие темпы создания национальной коллекции электронных карт;

в-третьих, реализация распределенных геоинформационных систем на уровне принятия решений при управлении кораблем.

Анализ тенденций, перспектив и направлений развития систем управления морскими динамическими объектами свидетельствует, что в результате развития вычислительной техники, способной перерабатывать огромные объёмы пространственной и иной информации геоинформационные системы будут внедряться на корабли, как базовый уровень развития ГИС.

Таким образом, применение геоинформационных систем в управлении кораблем является одной из **актуальных научных задач**. Это обусловлено динамической природой корабля, как объекта управления, для которого необходимо постоянное определение своего местоположения относительно других объектов, прокладывание маршрутов, зачастую в сложных метеорологических условиях, а также интегрирование атрибутивных (некоординатных) характеристик корабельных систем в управленческие решения.

В проблеме построения ГИС корабля можно выделить следующие актуальные направления:

1. Развитие устройств преобразования электромагнитной, акустической энергии во всех диапазонах спектра за счет применения новых материалов, форм построения и технологий производства.

2. Развитие устройств генерирования и приема электромагнитных сигналов во всех диапазонах электромагнитного спектра за счет применения новой элементной базой и более совершенных схем их построения.

3. Развитие методов и способов обработки электромагнитных и акустических сигналов.

4. Развитие средств и прикладных платформ обработки сигналов и данных.

5. Развитие средств визуализации данных на основе применения новых технологий отображения информации, мультимедийных форм ее представления, а также за счет использования новой элементной базы.

6. Интеграция радиоэлектронных систем в единую информационную инфраструктуру корабля с переходом к новой архитектуре их построения.

Первые четыре направления относятся к числу традиционных, хорошо освоенных и постоянно совершенствующихся технологий. В пятом направлении наметились за последние десять лет существенные позитивные сдвиги, в основном благодаря появлению новых информационных технологий. [72,76]

Шестое направление является относительно новым и потому перспективным. Новизна предлагаемых решений состоит в реализации возможностей базовых информационных технологий и поддерживающих их технических и программных средств посредством интеграции аппаратного обеспечения корабля в единую информационную пространственно распределенную инфраструктуру принятия решений.

Интеграция в предлагаемом смысле – это сложная комплексная задача, которая заключается:

- в переходе от функционально-специализированных архитектур средств обработки сигналов и данных к распределенной сетевой обработке; [76]

- во введении комплекса серверов обработки сигналов и данных в состав технического обеспечения; [76]

- в обеспечении взаимодействия комплексов, станций, систем и функциональных элементов не на основе парных связей, а на основе базовой информационной распределенной транспортной сети корабля; [76]

- в обеспечении развития базовой корабельной сети путем реализации возможности подключения к ней комплексов, станций, систем и функциональных элементов. [76]

Очевидно, что от правильного выбора технических средств и способов их интеграции зависит эффективность выполнения функциональных возможностей ГИС и полнота информационного обеспечения.

Под эффективностью ГИС понимается уровень решения ее основных задач с учетом затрат ресурсов и времени. Решение задачи эффективного ис-

пользования ГИС в судовождении сопряжено с рядом проблем, среди которых основными являются следующие:

- необходимость оперировать большими объемами разнородных геоданных, поступающих от разных источников и зачастую в несовместимых форматах;
- ограниченность пространства для размещения инфраструктуры ГИС на корабле;
- энергетические ограничения оборудования ГИС;
- отсутствие комплексного подхода проектирования ГИС корабля с учетом существующих ограничений на время распространения информации, требуемого уровня надежности доставки данных, инфраструктурных ограничений.

Перечисленные проблемы обуславливают актуальность задачи проектирования ГИС корабля под цели управления и выработку требований к характеристикам процессов обработки, хранения и передачи геоданных. Решение этой задачи представляет собой инженерную методику, обеспечивающую направленное формирование облика ГИС с заданным набором свойств. Определение заданного набора свойств потребует соответствующих моделей и расчетных методик оценивания характеристик ГИС.

Трудности проектирования ГИС корабля связаны с практическим отсутствием возможности использования физических моделей и натурального эксперимента. Поэтому при проектировании ГИС корабля особое значение уделяется этапу моделирования и эксперимента на модели. Моделирование позволит оценить характеристики исследуемого объекта, то есть решать прямую задачу моделирования. При наличии ограничений на характеристики объекта моделирование позволит настраивать параметры, которыми представлен объект, то есть решать обратную задачу моделирования. Модельный эксперимент позволит определить аналитические зависимости между настраиваемыми параметрами и характеристиками.

По вопросам моделирования функционирования ГИС и ее элементов, таких как системы хранения данных, телекоммуникационные сети, базы данных и соответствующих им процессов, таких как обработка и передача сигналов и данных опубликовано большое число работ. Среди многих отметим работы Т.И. Алиева, Ю.И. Рыжикова, П.П. Бескида, Е.П. Истомина, Л. Клейнрока, О.И. Кутузова, Т.М. Татарниковой, К. Шеннона, И.К. Лурье, В. Столлингса, А.И. Яшина. Работы этих и ряда других ученых составляют теоретическую базу моделирования процессов функционирования ГИС и ее элементов.

Объектом исследования является геоинформационная система морского динамического объекта, построенная на основе транспортной информационной сети корабля.

Предметами исследования являются аналитические и аналитико-статистические имитационные модели и методики структурно-функционального анализа проектных вариантов ГИС корабля.

Цель диссертации - теоретическое обоснование, разработка и исследование методов, моделей и алгоритмов расчета функциональных характеристик ГИС корабля для создания на этой основе методики структурного синтеза ГИС корабля с заданным набором качеств.

Для достижения поставленной цели в работе решены следующие **задачи**:

1. Разработана концептуальная модель ГИС корабля, как морского динамического объекта в геоинформационном пространстве, представленная на уровне информационных ресурсов, программного обеспечения и технических средств, отражающих функциональное назначение ГИС и ее архитектуру.

2. Разработана система математических моделей по оценке времени передачи данных в ГИС корабля для всех этапов информационного взаимодействия.

3. Разработана методика проектирования структурно-функциональной модели ГИС корабля, основанная на последовательном приближении ГИС к заданному набору свойств.

4. Выполнена программная реализация экспертной системы по выбору структурно-функциональной модели ГИС корабля.

Методы исследования. Математической основой моделирования функционирования ГИС и ее элементов служат теория систем и сети систем массового обслуживания, а также алгоритмические методы имитационного моделирования. Использование имитационного моделирования вызвано необходимостью учета динамических и стохастических характеристик морского объекта. Возникающие при этом задачи приводят к моделям, в которых критерии и ограничения, накладываемые на параметры ГИС, задаются не аналитически. Решение задач такого типа применительно к системам со значительным числом варьируемых параметров исключительно сложно и на практике часто сводится к многократно повторяющимся циклам моделирования, анализа и оценки полученных данных, корректировки параметров.

Решение сформулированной в диссертации проблемы проектирования инфраструктуры ГИС корабля базируется на методах системного анализа, теории вероятности, случайных процессов и математической статистики, систем и сетей систем массового обслуживания, методах численного анализа, теории оптимизации, имитационного моделирования.

Основные новые результаты, полученные в работе и выносимые на защиту:

1. Концептуальная модель ГИС корабля;
2. Система математических моделей оценки времени передачи данных в ГИС корабля;
3. Методика проектирования структурно-функциональной модели ГИС корабля;
4. Экспертная система автоматизированного выбора структурно-функциональной модели ГИС корабля.

Научная новизна.

Концептуальная модель ГИС корабля отличается описанием иерархии компонентов, поддерживающих функциональность ГИС, что позволяет выполнить структурную оптимизацию ГИС корабля под цели плавания.

Система математических моделей отличается сочетанием аналитического и статистического методов моделирования на всех этапах прохождения сигнала, включая установление соединения и повторную передачу в случае ошибки, что позволяет точнее оценить время передачи данных в условиях близких к реальным.

Методика проектирования структурно-функциональной модели ГИС корабля отличается комбинированным применением автоматической генерации вариантов ГИС и экспертных данных по выбору моделей ее построения, что позволяет осуществлять многокритериальную процедуру проектирования ГИС корабля.

Экспертная система, реализующая методику проектирования структурно-функциональной модели ГИС корабля, отличается применением сценарного подхода, что позволяет получить варианты инфраструктурных решений с учетом существующих модулей построения ГИС.

Практическая ценность и реализация результатов работы. Практическая ценность результатов диссертации заключается в полученных расчетных выражениях, алгоритмах и методиках, реализующих проектирование инфраструктуры ГИС корабля, которые могут быть рекомендованы для использования при проектировании интегрированных систем управления кораблем. Ценность результатов подтверждается авторскими свидетельствами, полученными в процессе исследований:

авторское свидетельство № 2016611252 «Программа оптимизации структуры защищенной компьютерной сети с применением генетического алгоритма»;

авторское свидетельство №2016611251 «Экспертная система выбора оптимальных средств защиты электронного контента».

Основные научные результаты диссертации используются в учебном процессе на кафедре при изучении дисциплин "Моделирование морских информационных систем" для студентов специальности 17.03.01.

Апробация работы. Основные научные результаты диссертации докладывались, обсуждались и были одобрены на VII-й Санкт-Петербургской межрегиональной конференции «Информационная безопасность регионов России», Санкт-Петербург, 26-28 октября 2011; 67-й научно-технической конференции, посвященной дню радио, Санкт-Петербург, май 2012; XIV-й Санкт-Петербургской международной конференции «Региональная информатика «РИ-2014», Санкт-Петербург, 29-31 октября 2014; IV-й Международной НПК «Информационные технологии в сервисе» ITS-2014, 18-19 декабря 2014; международной научно-практической конференции «Наука и образование в XXI веке», Тамбов, 31 января 2012 и 2015, кафедральных семинарах (РГГМУ 2011-2016 гг.).

Публикации. Основное содержание диссертации опубликовано в 15 печатных работах, в том числе в трех работах, рекомендованных ВАК РФ.

Структура и объем работы. Диссертация состоит из введения, трех глав, заключения и списка литературы (всего 79 источник) Общий объем работы – 128 страницы сквозной нумерации, в том числе основного текста – 103 машинописных страниц, 43 рисунков и графиков, 4 таблиц.

Во введении раскрывается актуальность решения научной задачи, сформулированы объект, предмет, цели и задачи исследования, показана теоретическая и практическая значимость работы, приведено краткое содержание работы по разделам, и перечислены основные научные результаты, выносимые на защиту.

В первой главе «Анализ элементов геоинформационной системы поддержки принятия решения при управлении морским динамическим объектом» дается характеристика корабля как динамическому объекту в геоинформационном пространстве, определяются основные элементы геоинформационного пространства, среди которых определена важность использования электрон-

ных карт для повышения безопасности мореплавания и обеспечения своевременного принятия обоснованных решений судоводителем при управлении кораблем.

Предложена концептуальная модель распределенной ГИС корабля, представленная в виде трехслойной структуры: внутренний слой, соответствует информационному обеспечению, средний – программному обеспечению и внешний – аппаратному обеспечению. Подробно рассматривается обеспечение для каждого слоя.

На основе концептуальной модели предложена структурно-функциональную модель, которая детализирует процесс взаимодействия лица принимающего решение (ЛПР) при управлении кораблем с серверной частью ГИС.

На основе структурно-функциональной модели ГИС морского морского динамического объекта сформулирована задача диссертации, как задача разработки методического обеспечения формирования ГИС корабля под заданные цели плавания и учетом ограничений на требуемые показатели производительности ГИС при работе с актуальными данными и занимаемые площади.

Результат, выносимый на защиту - концептуальная модель ГИС корабля, представленная на уровне информационных ресурсов, программного обеспечения и технических средств, отражающих функциональное назначение ГИС и ее архитектуру.

Во второй главе «Система моделей оценки производительности геоинформационной системы поддержки принятия решений при управлении кораблем» согласно контексту связи лица принимающего решение с серверной частью ГИС предложены следующие математические модели:

M1 – модель установления соединения в распределенной ГИС корабля;

M2 – модель оценки времени передачи данных клиенту ГИС от момента получения запроса;

M3 – модель оценки времени решения задачи

Система моделей охватывает все этапы информационного взаимодействия «Клиент-Сервер» и позволяют оценить время прохождения геоинформации по всем функциональным модулям.

Суть модели М1 заключается в том, что случайным образом разыгрываются неработоспособные узлы и транзиты распределенной ГИС и определяется время установления соединения в заданных условиях. Модель позволяет оценить важные параметры, такие как количество альтернативных маршрутов, допустимое число неработоспособных коммутаторов и транзитов, необходимое количество попыток для повторной передачи, чтобы сохранить живучесть сети, то есть ее работоспособность.

Исходными данными в модели М1 являются структурная и потоковая метрики и ограничения, при которых должно выполняться моделирование. Результатом моделирования являются статистики, позволяющие оценить вероятность установления соединения за время, не превышающее допустимое, средние и среднеквадратические оценки значения времени установления соединения.

Оценка времени передачи данных в ГИС корабля (модель М2) решена с применением аппарата теории очередей и преобразования Лапласа-Стилтьеса

Исходными данными в модели М2 являются множество транзитов маршрута и мощность этого множества. Результатом является определение «узкого» места маршрута, то есть этапа или этапов информационного взаимодействия, вносящих наибольшую задержку в значение времени передачи данных.

Математической базой для оценки время решения задачи в АРМ (модель М3) выбраны разомкнутые сети массового обслуживания

Исходными данными в модели М3 являются характеристики производительности процессоров, трудоемкость обслуживания отдельного решения, число обращений к внешнему запоминающему устройству, интенсивность обслуживания заданий.

Модель М3 позволяет найти схему комплексирования клиентской части ГИС с серверной.

Результат, выносимый на защиту - система математических моделей по оценке производительности ГИС корабля для всех этапов информационного взаимодействия.

В третьей главе «Методика проектирования структурно-функциональной модели геоинформационной системы корабля» предложена оригинальная методика формирования инфраструктуры ГИС поддержки принятия решения при управлении кораблем, основанная на последовательном приближении ГИС к заданному набору свойств.

Методика включает в себя, во-первых, алгоритм решения, обеспечивающий формирование инфраструктуры ГИС и во-вторых, приближение архитектуры ГИС к заданному набору свойств.

Алгоритм формирования инфраструктуры ГИС состоит из следующих действий:

1. Определение исходных данных для построения ГИС.
2. Оценка временных характеристик.
3. Определение полного списочного состава элементов с техническими характеристиками, на которых строится ГИС, таких как канал связи и АРМы (процессор, жесткий диск, оперативная память) удовлетворяющего требованию производительности ГИС.
4. Определение “узкого” места в структуре ГИС.

Включает последовательность пошаговых действий для приближения архитектуры ГИС к заданному набору свойств.

Результат, выносимый на защиту - методика проектирования структурно-функциональной модели ГИС корабля, основанную на последовательном приближении архитектуры ГИС к заданному набору свойств.

В четвертой главе «Экспертная система по выбору облика геоинформационной системы поддержки принятия решений при управлении кораблем» раскрываются особенности программной реализации экспертной системы, которая автоматизирует методику проектирования структурно-функциональной модели ГИС корабля.

Выбор варианта инфраструктурного решения построения ГИС основан на сценарном подходе и алгоритме «прямой волны», согласно которому поиск решения идет от исходных данных к целевому параметру.

Экспертная система построена по модульному принципу и состоит из следующих компонентов: рабочей памяти, называемой также базой данных, базы знаний, решателя, подсистем приобретения знаний, объяснений и диалога.

Результат, выносимый на защиту - экспертная система по выбору облика ГИС корабля, основанной на предложенной методике проектирования ГИС корабля.

В дальнейших исследованиях предлагается развитие методов, моделей и технологий построения геоинформационной системы поддержки принятия решения при управлении морским динамическим объектом, для совершенствования процесса управления морским динамическим объектом и повышения качества геоинформационного обеспечения в задачах повышения безопасности мореплавания.

1. Анализ элементов геоинформационной системы поддержки принятия решения при управлении морским динамическим объектом

1.1 Корабль как морской динамический объект

К морским динамическим объектам (МДО) относятся наиболее распространенные водоизмещающие суда, корабли, суда с динамическим принципом поддержания (на воздушной подушке, на подводных крыльях), подводные аппараты-роботы, поисково-разведочные комплексы и др.

Морские динамические объекты МДО характеризуются изменением в пространстве и времени, а так же определением своего положения относительно других объектов пространства. При определении местоположения МДО в пространстве необходимо иметь три координаты (плоские координаты X , Y и высоту H) и точное определение времени распространения сигнала. Поэтому в GPS-приемнике измеряется расстояние до четырех различных искусственных спутников Земли (ИСЗ).

Поскольку морской динамический объект характеризуется изменением в пространстве за определенный промежуток времени, то его можно причислить к объектам геоинформационного пространства.

Объекты геоинформационного пространства описываются пространственными, временными и тематическими характеристиками.

Пространственные характеристики определяют положение объекта в заранее определенной системе координат, основное требование к таким данным – точность.

Временные характеристики фиксируют время исследования объекта и важны для оценки изменений свойств объекта с течением времени. Основное требование к таким данным – актуальность, что означает возможность их использования для обработки, неактуальные данные – это устаревшие данные.

Тематические характеристики описывают разные свойства объекта, включая экономические, статистические, технические и другие свойства, основное требование – полнота описания объекта.

Для представления пространственных объектов в ГИС используют пространственные и атрибутивные типы данных.

Пространственные данные – сведения, которые характеризуют местоположение объектов в пространстве относительно друг друга и их геометрию.

Пространственные объекты представляют с помощью следующих графических объектов: точки, линии, области и поверхности.

Описание объектов осуществляется путем указания координат объектов и составляющих их частей.

Точечные объекты – это такие объекты, каждый из которых расположен только в одной точке пространства, представленной парой координат X , Y . В зависимости от масштаба картографирования, в качестве таких объектов могут рассматриваться дерево, дом или город.

Линейные объекты, представлены как одномерные, имеющие одну размерность – длину, ширина объекта не выражается в данном масштабе или не существенна. Примеры таких объектов: реки, границы муниципальных округов, горизонтالي рельефа.

Области (полигоны) – площадные объекты, представляются набором пар координат (X , Y) или набором объектов типа линия, представляющих собой замкнутый контур. Такими объектами могут быть представлены территории, занимаемые определенным ландшафтом, городом или целым континентом.

Поверхность – при ее описании требуется добавление к площадным объектам значений высоты. Восстановление поверхностей осуществляется с помощью использования математических алгоритмов (интерполяции и аппроксимации) по исходному набору координат X , Y , Z .

Общее цифровое описание пространственного объекта включает: наименование; указание местоположения; набор свойств; отношения с другими

объектами. Наименованием объекта служит его географическое название (если оно есть), его условный код или идентификатор, присваиваемый пользователем или системой.

Однотипные объекты по пространственному и тематическому признакам объединяются в слои цифровой карты, которые рассматриваются как отдельные информационные единицы, при этом существует возможность совмещения всей имеющейся информации [19,20,77]

Для представления пространственных данных в ГИС применяют векторные и растровые структуры данных.

Векторная структура – это представление пространственных объектов в виде набора координатных пар (векторов), описывающих геометрию объектов. Векторное представление пространственных данных приведено на рисунке 1.1.

Растровая структура данных предполагает представления данных в виде двумерной сетки, каждая ячейка которой содержит только одно значение, характеризующее объект, соответствующий ячейке растра на местности или на изображении. В качестве такой характеристики может быть код объекта (лес, луг и т.д.) высота или оптическая плотность.

Точность растровых данных ограничивается размером ячейки. Такие структуры являются удобным средством анализа и визуализации разного рода информации (рисунок 1.2).

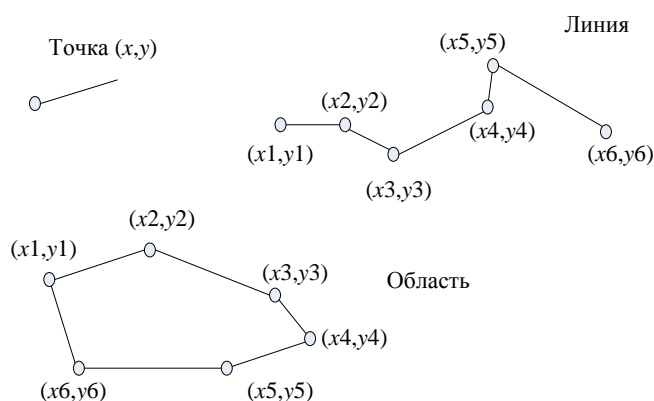


Рисунок 1.1 – Векторное представление пространственных данных

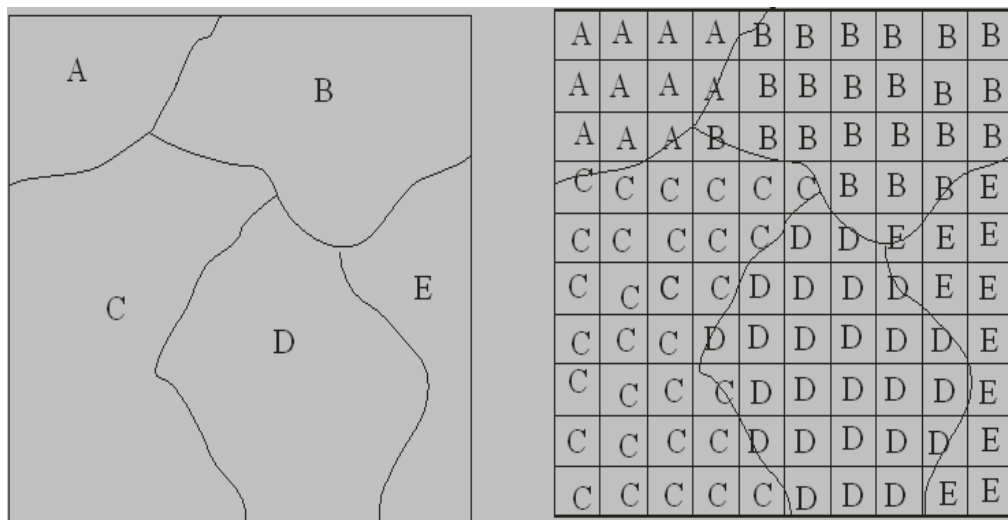


Рисунок 1.2 – Растровая структура данных

Для повышения безопасности мореплавания и обеспечения возможности быстрого принятия правильных и обоснованных решений судоводителю в наглядном и легко интерпретируемом виде должна представляться информация, характеризующая все стороны процесса судовождения. Основой такой интегрированной информации является электронная карта.

В настоящее время все электронные карты, которые производятся и предлагаются пользователям для управления кораблем, можно разделить на ряд категорий по типу, формату, соответствию спецификации на их изготовление и легитимности издания (таблица 1.1).

Таблица 1.1– Категории электронных карт

Категория	Тип	Формат	Спецификация	Изготовитель
А	Векторные	ИНО S-57(e3)	S-57 enc prod spec	Государств. ГС
В	Векторные	ИНО S-57(e3)	S-57 enc prod spec	Частные ГС
С	Векторные	ИНО S-57 v2-e3	Не соответствует S-57 enc prod spec	Государств. ГС
Д	Векторные	ИНО S-57 v2-e3	Не соответствует S-57 enc prod spec	Частные ГС
Е	Векторные	Другие	Не соответствует S-57 enc prod spec	Государств. ГС
Ф	Векторные	Другие	Не соответствует S-57 enc prod spec	Частные ГС
Г	Растровые	Нескольких ГС	Не соответствует между-народному стандарту	Государств. ГС
Н	Растровые	Различные	Не соответствует между-народному стандарту	Частные ГС

Основные документы, регламентирующие электронные навигационные карты:

– Резолюция ИМО А.817(19) «Эксплуатационные требования к электронным картографическим навигационным информационным системам (ECDIS)»;

– Стандарт МГО S-57 «Стандарт для обмена цифровыми гидрографическими данными»;

– Стандарт МГО S-52 «Стандарт содержания карт и отображения в ECDIS»[1,6,12]

1.2 Характеристика компонентов ГИС корабля

Большинство современных ГИС осуществляют комплексную обработку информации, к функциям которой относятся:

1. Ввод и редактирование данных;
2. Поддержка моделей пространственных данных;
3. Хранение информации;
4. Преобразование систем координат и трансформация картографических проекций;
5. Растрово-векторные операции;
6. Измерительные операции;
7. Полигональные операции;
8. Операции пространственного анализа;
9. Различные виды пространственного моделирования;
10. Цифровое моделирование рельефа и анализ поверхностей;
11. Вывод результатов в разных формах.[20]

Комплексный характер функций ГИС поддерживается соответствующими компонентами, которые образуют трехслойную структуру: внутренний слой соответствует информационному обеспечению ГИС, средний – программному обеспечению ГИС и внешний – это слой аппаратного обеспечения ГИС (рисунок 1.13).

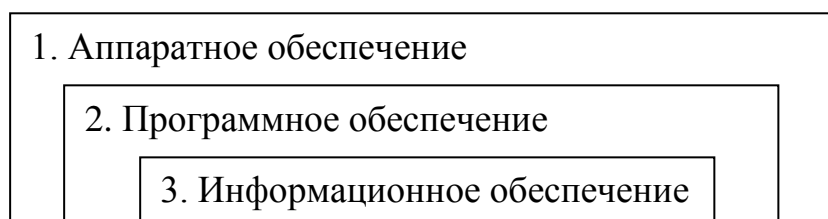


Рисунок 1.3 – Компоненты ГИС

Информационное обеспечение включает в себя все картографические данные, которые образуются в ГИС за время плавания, а также любую другую информацию, циркулирующую в ГИС и необходимую для управления про-

цессами обработки, хранения и передачи данных на корабле и при взаимодействии с внешней средой.

Программное обеспечение реализует функциональные возможности ГИС и состоит из базового программного обеспечения, такого как операционные системы, системы управления базами данных, системы визуализации данных и других и прикладного программного обеспечения, предназначенного для решения специализированных задач судовождения, обработки сигналов, обработки и передачи данных и других.

Аппаратное обеспечение – это комплекс технических средств, применяемых при функционировании ГИС. К ним относятся рабочие станции, устройства ввода-вывода информации, устройства обработки и хранения данных, средства телекоммуникаций.

1.3 Информационное обеспечение ГИС корабля

Информационное обеспечение ГИС корабля составляют навигационные карты и данные, необходимые для управления объектом, вместе они образуют электронные картографические навигационные информационные системы ЭКНИС (ECDIS – Electronic Chart Display and Information System).

ЭКНИС осуществляет отображение карт и место корабля, позволяет вести прокладку трассы движения и осуществлять контроль отклонений от заданного маршрута, осуществляет вычисление безопасных курсов, предупреждение судоводителя об опасности, ведение судового журнала, управление авто-рулевым и т.п.

Все существующие электронные картографические системы принято делить на три группы:

- ECDIS – электронные картографические навигационные информационные системы;
- ECS – электронные картографические системы;
- RCDS – растровые картографические дисплейные системы.

Характеристики, отличающие эти группы электронно-картографических систем, приведены на рисунке 1.4.

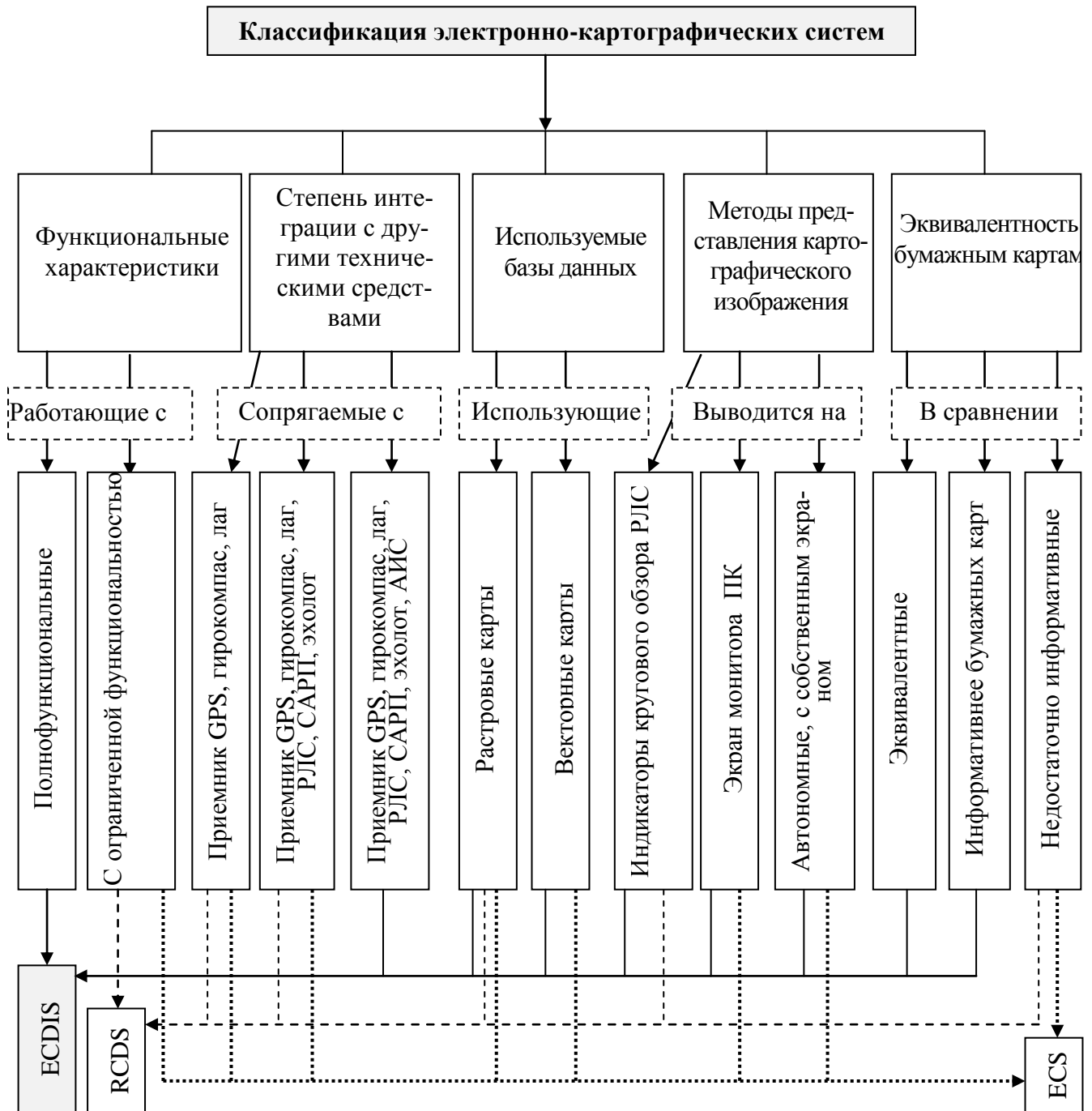


Рисунок 1.4 – Классификация электронно-картографических систем

Международной морской организацией (ММО) официально признаются только ECDIS, которые в отношении оборудования и базы данных самой электронной карты должны в обязательном порядке соответствовать строгим спецификациям и стандартам, выработанным ММО и Международной гидрогра-

фической организацией. Имеется в виду, что с юридической точки зрения ECDIS является эквивалентом современных бумажных навигационных карт в рамках требований 20-го правила главы V безопасности мореплавания международной конвенции по охране человеческой жизни на море, известная как Конвенции SOLAS.[27]

В группу ECS включаются все остальные системы, т. е., которые не соответствуют стандартам ММО и потому не приняты в качестве эквивалента бумажным картам. Поэтому, несмотря на наличие ECS на борту корабля, бумажные карты по-прежнему должны использоваться судоводителями для ведения навигационной прокладки. Электронная картографическая система такого типа рассматривается лишь как средство, дублирующее прокладку на бумажных морских навигационных картах.

Группа ECS весьма многочисленна и отличается большим разнообразием типов и версий входящих в нее систем.

В растровых электронных картографических системах в качестве основного элемента используются электронные карты растровых форматов. Эти карты получают путем сканирования бумажных навигационных карт. В 1998 г. RCDS выделены в отдельную группу. ММО признала возможным их использование в практике судовождения до появления в достаточном количестве конвенционных векторных карт.

В настоящее время процесс внедрения сертифицированных электронно-картографических систем приобрел системный характер. Причем, как для морских, так и для речных систем. Международная морская организация ввела требования по использованию ECDIS систем на морских судах. Процесс обязательного внедрения будет проходить поэтапно, начиная с 2012 года. К 2018 году морские суда с валовой вместимостью от 500 т. и выше должны быть оснащены такими системами

Электронная картографическая навигационная информационная система отображает на экране дисплея точные картографические данные морской карты в реальном времени, т. е. в сочетании с текущим местом объекта, получен-

ным от ГЛОНАСС и GPS. Система обрабатывает и представляет информацию и от другого навигационного оборудования, например гироскопа, лага, эхолота, радиолокационной станции (РЛС), системы автоматической радиолокационной прокладки (САРП) и автоматической идентификационной системы (АИС). На рисунке 1.5 показаны основные элементы ECDIS.[]

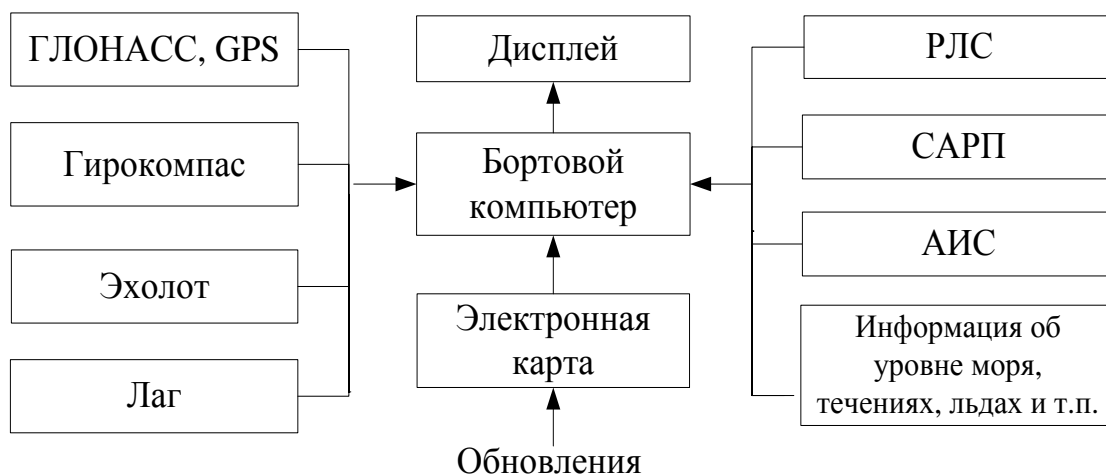


Рисунок. 1.5 – Основные элементы ECDIS

К информационному обеспечению, необходимому для управления кораблем также относят гидрологические данные, данные о погодных явлениях, ледовой обстановке. Автоматический комплексный сбор и обработку этих данных выполняет бортовой компьютер.

Данные об окружающей обстановке в радиусе несколько десятков км обрабатывает и передает на бортовой компьютер автоматическая идентификационная система (АИС).

Действие АИС основано на приеме и передачи сообщений по УКВ волнам по протоколу самоорганизации множественного доступа с разделением времени SOTDMA. Согласно этому протоколу кадр (frame) делится на 2500 временных слотов, в одном из которых в течение 1-й минуты регистрируется входящее в зону радиовидимости (рисунок 1.6). Таким образом, предельное количество объектов, контролируемых АИС корабля - 2500.

АИС включает в себя следующие компоненты: транспондер, состоящий из УКВ передатчика и одного - двух УКВ приемников, приемник глобальной спутниковой навигации, модулятор/демодулятор, контроллер на основе микропроцессора, оборудование ввода-вывода информации на экран бортового компьютера или другие элементы управления.

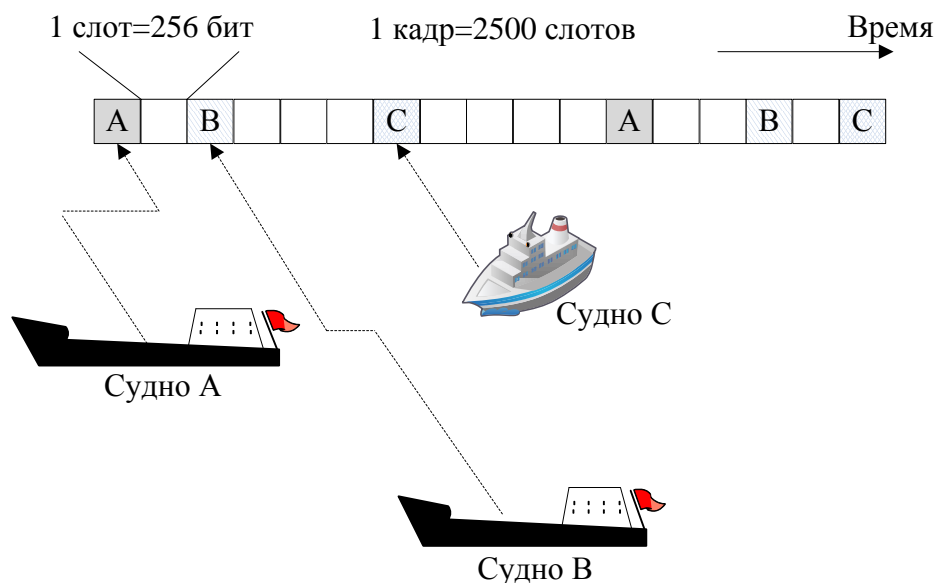


Рисунок 1.6 – Принцип работы АИС корабля

Передатчик АИС работает на более длинных волнах, чем радары, что позволяет производить обмен информацией не только на прямых расстояниях, но и местности, имеющей препятствия в виде не очень больших объектов, а также при плохих погодных условиях. Хотя достаточно одного радио канала, некоторые АИС системы передают и получают по двум радио каналам для того, чтобы избежать проблем интерференции и не нарушать коммуникацию других объектов. [2,7,36,40,41]

До появления в зоне видимости транспондера других судов (целей) аппаратура АИС формирует координаты места и ведет синхронизацию времени с помощью системы глобальной спутниковой навигации.

При появлении в зоне видимости транспондера других целей процессор транспондера, сканируя слоты кадра в минутном интервале, определяет свободный слот, который занимает для собственной идентификации в диапазоне кадра. Кроме того, на транспондер поступают данные с периферических приборов – гироскопа, лага и пр., эти данные кодируются и посредством двухканальной УКВ-радиостанции передаются в эфир, вместе с дополнительными данными, например, сводками погоды, своих намерений, и т. д., которые могут набираться при необходимости с клавиатуры. Частота "выбрасывания" в эфир таких пакетов регламентирована Резолюцией ИМО и зависит от категории, к которой относится содержащаяся в пакете информация, и скорости объекта. Так, например, динамическая информация при скорости движения 26 км/ч должна передаваться через каждые 2 секунды.[]

После этого транспондер принимает из эфира динамическую, рейсовую и статическую информацию от других судов, находящихся в зоне видимости, с транспондеров, установленных на этих судах. Эта информация отображается по заданному критерию на мониторе электронной картографии ECDIS в виде названия (идентификатора), позиции, вектора скорости и т. д., причем позиция, курс и размеры судов - графически в виде ориентированного контура объекта.[3,4]

Передаваемую/принимаемую АИС информацию подразделяются на 4 категории – статическую, динамическую, о рейсе, относящуюся к безопасности.

Статичная информация

- Номер MMSI
- Номер Международной морской организации (ИМО)
- Радиопозывной и название плавучего средства
- Габариты
- Тип плавучего средства
- Данные о месте антенны

Все статические данные вводятся в бортовой компьютер при установке

оборудования. Частота передачи статических данных в эфир – каждые 6 мин.

Динамическая информация

- Местоположение (широта и долгота)
- Возраст информации (как давно обновлялась)
- Курс
- Скорость
- Угловая скорость поворота
- Статус (к примеру: Не могу управляться или Ограничен в возможности маневрировать).[24]

Частота передачи динамической информации зависит от скорости корабля и изменения его курса от 2 с до 3 мин. Если объект находится на якоре или движется с малой скоростью, то интервал между сообщениями динамической информации составляет 3 мин. При повышении скорости, то интенсивность передач увеличивается. При скорости судна 23 узла и более интервал времени между соседними передачами динамической информации составляет всего 2 с. Такая адаптация интервала передач к динамике объекта позволяет отследить перемещение и все маневры, и в то же время не перегружать эфир излишними передачами при медленном перемещении корабля.[24]

Рейсовая информация

- Пункт назначения
- Время прибытия
- Наличие (тип) опасного груза
- Осадка судна

Эти данные передаются каждые 6 мин., при изменении данных и по требованию.

Информация по безопасности представляет собой короткие текстовые сообщения в свободном формате с использованием кодировки ASCII. Они могут быть адресованы как конкретному объекту (или береговой станции), так и всем станциям. Передача этих сообщений осуществляется оператором путем набора текста на пульте управления и отображения информации на экране.[12]

Эти данные передаются по необходимости.

Также предусмотрена передача дополнительной текстовой информации между терминалами АИС (пейджинг). Передача такой информации возможна как в адрес всех терминалов в радиусе действия, так и одному определенному терминалу.[12]

1.4 Программное обеспечение ГИС корабля

Программный слой ГИС корабля образуют специализированные и базовые информационные технологии, реализующие процессы обработки, передачи и хранения информации в ГИС.

На рисунке 1.7 приведена структура программного слоя ГИС корабля. Информационные технологии протоколов обмена реализуют:

- поддержку протокола обмена по СНЧ, СДВ, СВ, КВ, УКВ, радиорелейным, космическим, гидроакустическим, проводным каналам;
- поддержку сетевых протоколов типа TCP/IP, ATM, X.25.
- поддержку стандартов защиты информации в каналах связи и сетях ГИС.[45,50,54,77]



Рисунок 1.7 – Структура программного слоя ГИС корабля

Информационные технологии обработки сигналов реализуют:

- обработку радиолокационных сигналов;
- обработку радиосигналов;
- обработку оптоэлектронных сигналов;
- обработку гидроакустических сигналов;
- оцифровка аналоговых сигналов. [69]

Информационные технологии обработки данных реализуют:

- обработку координат;
- прокладку маршрутов;
- учет целей;
- визуализацию данных;
- классификацию целей;
- обработку результатов измерений;
- выработку рекомендаций по маневрированию, счислению пути;
- ведение журналов учета. [69]

Базовые информационные технологии включают технологии:

- электронной почты;
- документооборота;
- СУБД;
- формирования, поиска и предоставления информационных ресурсов;
- защиты информационных ресурсов;
- электронной подписи;
- управления конфигурацией.

Состав информационных технологий зависит от полноты выполняемых функций ГИС корабля и может меняться в зависимости от целей плавания.

1.5 Аппаратное обеспечение ГИС корабля

К аппаратному слою ГИС корабля относят радиолокационные комплексы и станции, средства связи, гидроакустические комплексы и станции, аппаратуру опознавания, комплексы радионавигации, информационно-управляющие системы и другие средства, обеспечивающие преобразование и обработку электромагнитных (акустических) сигналов и данных от корабельных систем. [70]

Аппаратный слой ГИС корабля в общем случае составляют шесть функциональных модулей (ФМ). В каждый модуль входят устройства и средства, реализующую общую функцию:

ФМ1: устройства, осуществляющие преобразование электромагнитной (акустической) энергии, как в случае излучения, так и приеме сигналов, или другими словами - антенны.

ФМ2: устройства, осуществляющие прием, усиление, демодуляцию и декодирование сигналов – приемники.

ФМ3: устройства, осуществляющие прием, усиление, модуляцию и передачу сигналов – передатчики.

ФМ4: средства обработки сигналов и данных соответствующей прикладной платформой. Под прикладной платформой понимают структурированный набор информационных технологий и стандартов их реализации, обеспечивающей создание и применение средств обработки сигналов и данных от корабельных систем.

ФМ5: средства визуализации, обеспечивающие динамическое отображение целей и их траекторий движения; документирование данных на карту и бумажный носитель; просмотр документов и статистическое отображение наиболее важных данных.

ФМ6: устройства, обеспечивающие преобразование воздействий окружающей среды в электромагнитные сигналы – датчики. [70]

Архитектура аппаратного слоя с учетом взаимодействия функций приведена на рисунке 1.8.

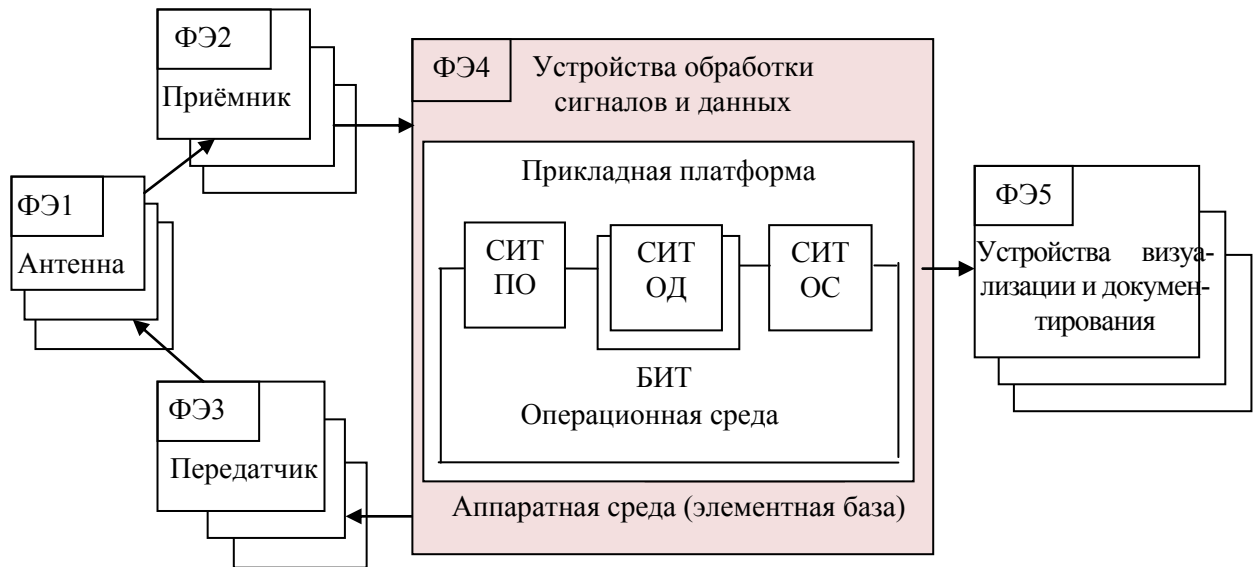


Рисунок 1.8 – Архитектура аппаратного слоя ГИС корабля

Изучение функций, методов приема и обработки сигналов в комплексах, станциях и образцах радиоэлектронного оборудования показывает, что между устройствами различных классов аппаратного слоя ГИС корабля существуют тесные информационные взаимосвязи. Результаты работы одних из них являются исходными данными для других. Возможность технической и информационной интеграции различных классов аппаратного обеспечения корабля реализуется третьим слоем ГИС корабля – его программным обеспечением.

1.6 Структурно-функциональная модель ГИС корабля

Структурно-функциональная модель ГИС корабля с учетом нового подхода представлена на рисунке 1.9 и представляет собой локальную вычислительную сеть (ЛВС) корабля с коммутацией сегментов. [70]

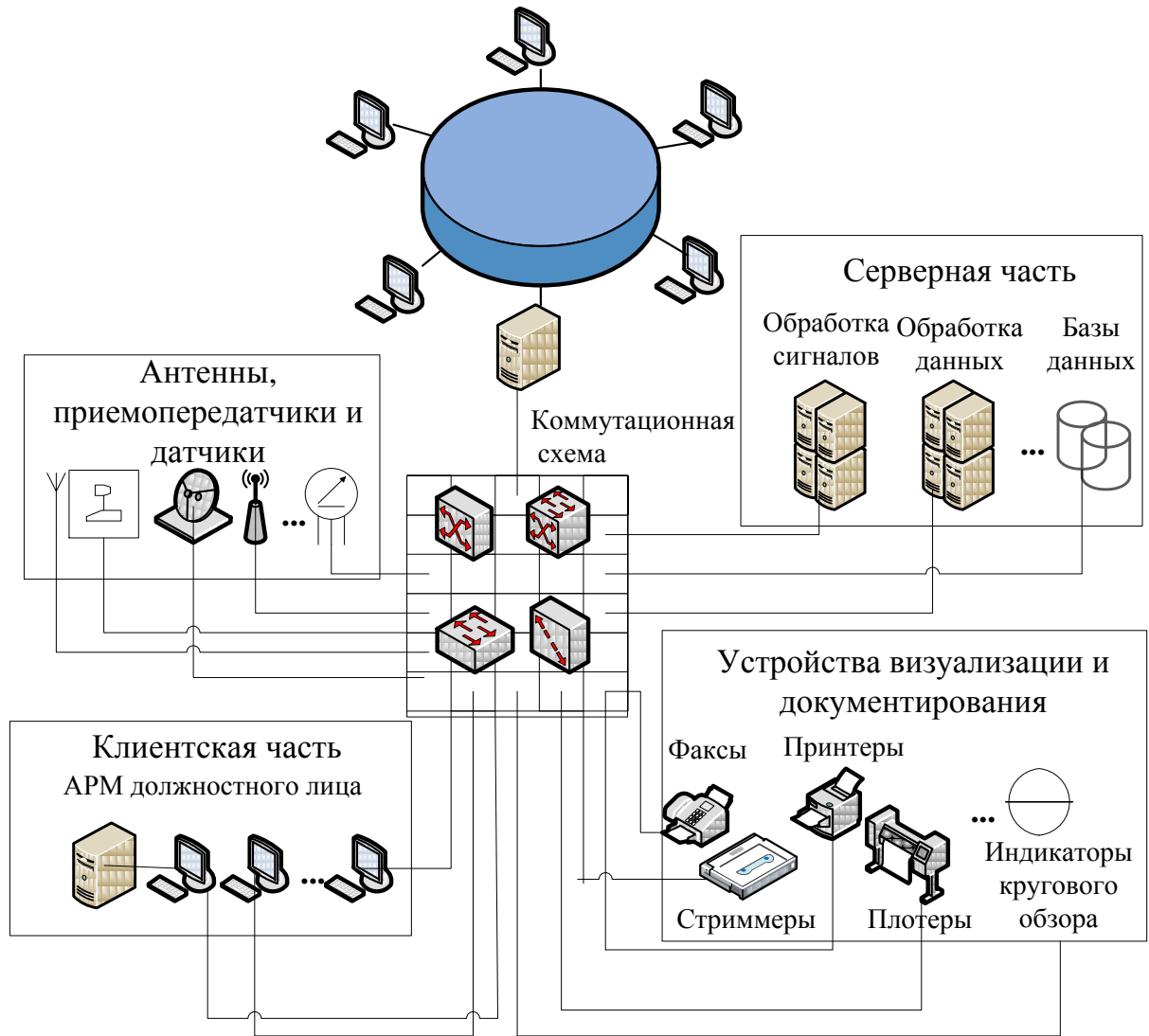


Рисунок 1.9 – Структурно-функциональная модель ГИС корабля

Технология коммутации сегментов основана на применении узлов коммутации (коммутаторов), позволяющих одновременно передавать данные между всеми взаимодействующими парами узлов ЛВС корабля взамен разделяемых линий связи между всеми узлами сегмента. Для коммутатора вся сеть представляется наборами MAC-адресов узлов, образующих ЛВС, то есть комплексов, станций, систем и средств, рассмотренных при описании аппаратного слоя ГИС корабля. [75]

Связь между узлами ЛВС корабля на физическом уровне осуществляется с помощью физической среды распространения сигнала – проводных и беспроводных линий связи, на логическом уровне – с помощью механизма сооб-

щений, при этом управляющие сообщения и данные, пересылаемые между узлами ЛВС, передаются в виде пакетов данных.

Процессы продвижения пакетов коммутаторами основаны на существовании хотя бы одного логического пути между любыми двумя узлами сети. [75]

Поскольку к ЛВС корабля предъявляются высокие требования по надежности доставке данных, вызванные обеспечением безопасности и продолжительностью плавания, то количество возможных путей доставки данных должно быть избыточным (рисунок 1.10). Это позволяет, несмотря на отказы отдельных элементов, сохранять работоспособность, т.е. обеспечивает живучесть ЛВС корабля.

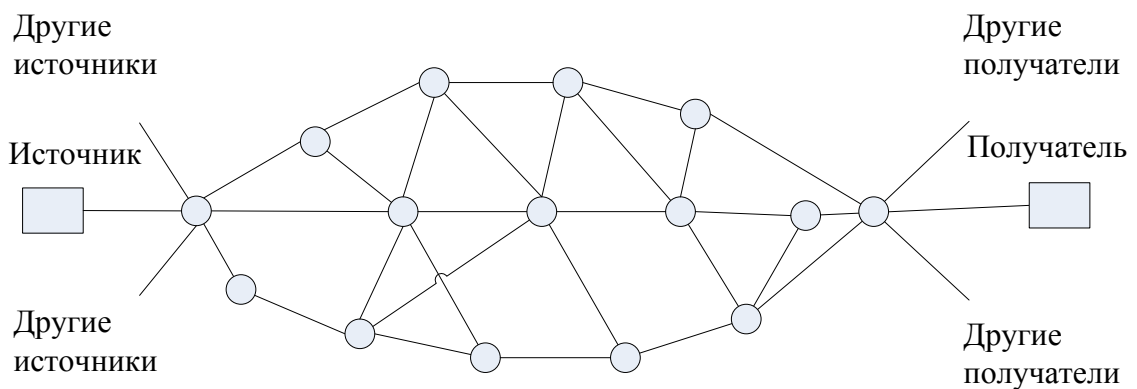


Рисунок 1.10 – Альтернативные пути доставки данных в ЛВС корабля

Доставка данных начинается с установления соединения и представляет собой посылку и прохождение вызова от источника к адресату по одному из множества альтернативных маршрутов. Каждый такой маршрут – это логический канал, построенный от источника к адресату для последующей по нему передачи данных. Логический канал состоит из транзитов (линий связи), связывающих узлы коммутации, входящих в данный маршрут. С точки зрения прохождения вызова от станции к станции состояние транзита бинарно: либо транзит занят, и вызов по нему не проходит (состояние транзита «1»), либо

транзит имеет канал требуемой скорости, и вызов через такой транзит пройдет (состояние транзита «0»).[78]

Схема взаимодействия клиентов ГИС с серверной частью ГИС корабля приведена на рисунке 1.11.

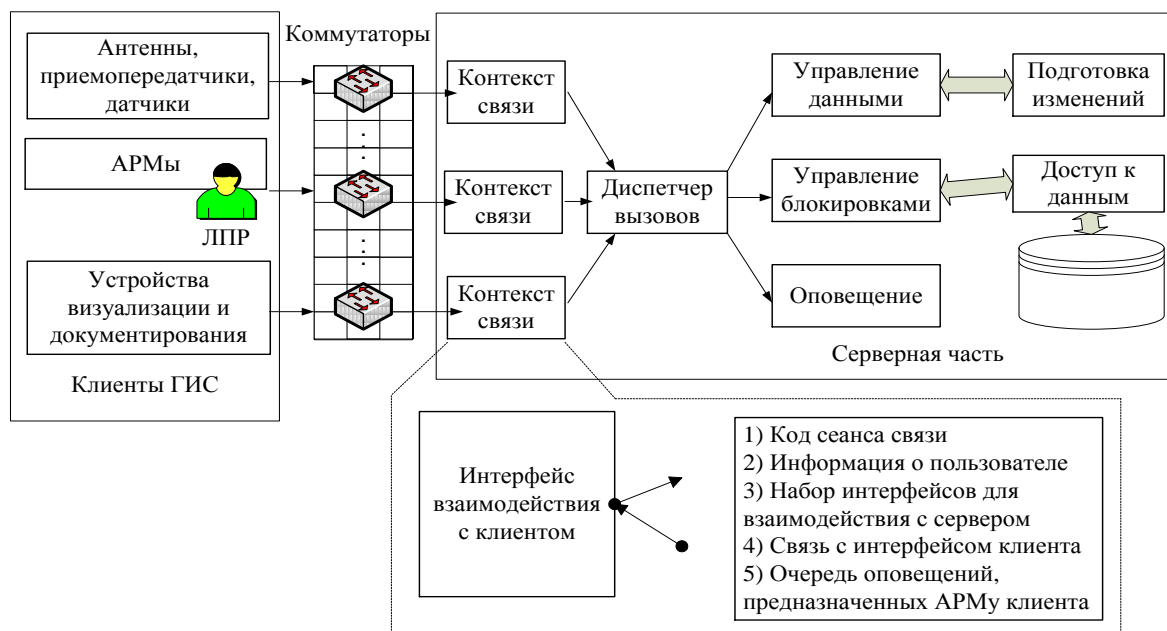


Рисунок 1.11 – Схема взаимодействия клиентов с серверной частью ГИС корабля

ГИС корабля должна обеспечивать возможность работы с одним набором геоинформационных данных одновременно нескольким клиентам, одним из них информация необходима для просмотра, другая для внесения изменений в пространственные данные. Если изменения вносятся со стороны одного из клиентов в общие данные, то для приведения информации, размещенной в оперативной памяти клиентской программы в актуальное состояние, система оповещения серверной части рассылает управляющую информацию о вносимых изменениях.

Также в работе ГИС должна быть предусмотрена возможность рестартов для повторных попыток установления соединений через случайное время time-

out, блокировка одновременного изменения одного объекта со стороны нескольких клиентов.

Все перечисленные функции ГИС корабля выполняются с применением технологии "клиент/сервер", согласно которой все задачи по обработке данных выполняются серверной частью, а клиентские приложения выполняют функции мониторов соответствующих АРМов специалистов. По своей сути приложение – это набор интерфейсов и адреса функций, предписанных АРМу.

Контекст связи предназначен для обеспечения взаимодействия клиента с той функцией серверной части, которая специфична для конкретного клиента (рисунок 1.12).

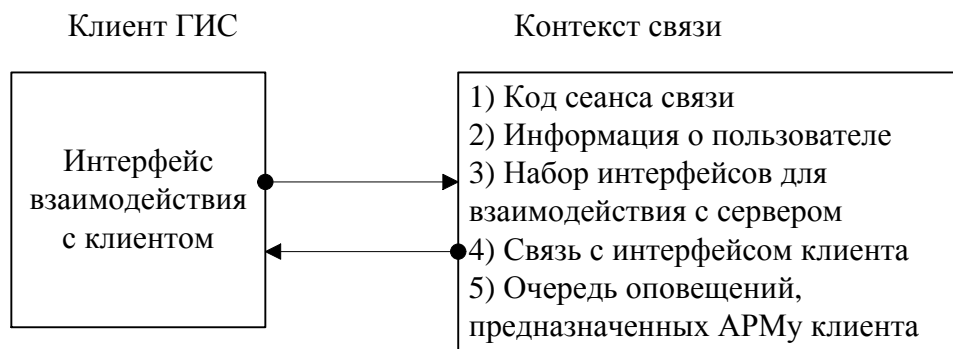


Рис. 1.12 – Контекст связи клиента ГИС

Диспетчер вызовов управляет очередью запросов функций сервера со стороны клиентов.

Управление данными – это функция серверной части по обработке запросов клиентов на получение и внесение изменений в пространственные данные.

Подготовка изменений – это функция серверной части, которая преобразует высокоуровневые запросы, например, изменение контура объекта в низкоуровневые операции изменения физических данных и помещает их во входную очередь подсистемы доступа к физическим данным, которая в фоновом режиме вносит изменения в физические данные.[78,79]

Управление блокировками – это функция, предназначенная для предотвращения одновременного внесения изменений в один объект со стороны нескольких клиентов.

Оповещение – это функция серверной части, предназначенная для формирования пакета оповещений (протокола изменений) в соответствии с которой копия этого пакета транслируется в очередь оповещений каждого контекста связи. О том, что поступили новые оповещения сервер сообщает клиентам посредством функций интерфейса взаимодействия с клиентом. При получении оповещений клиентами, сервер автоматически удаляет их из очереди. Клиенты, получая от сервера пакеты оповещений, хранящиеся в очереди контекста связи, обрабатывают полученные оповещения и приводят в актуальное состояние свои оперативные данные.

Система оповещений – это также инструмент администратора ГИС корабля, т.к. оповещения могут иметь управляющий характер и использоваться для автоматического завершения работы всех клиентов, присоединенных в данный момент к серверу данных или же для рассылки визуальных сообщений от администратора к пользователям ГИС и другое.[62]

1.7 Постановка задачи исследования

Сформулируем задачу диссертации, как задачу разработки инженерной методики проектирования ГИС корабля под заданные цели плавания и учетом ограничений на требуемые показатели производительности ГИС при работе с актуальными данными и занимаемые площади.

Инженерная методика должна содержать последовательность шагов, достаточных для получения структурно-функциональной модели ГИС корабля.

При решении поставленной задачи средствами моделирования будем оценивать производительность ГИС корабля. Производительность может характеризоваться пропускной способностью узлов и каналов связи ГИС и/или

вариациями времени доставки. Поскольку ГИС корабля – это интеграция неоднородного технического оборудования в единую информационную инфраструктуру корабля, то в качестве производительности в таком случае рекомендуется взять время доставки данных, т.к. пропускная способность ассоциируется с отдельным узлом, а не с сетью, какой является ГИС корабля. Это требует моделей и расчетных методик, учитывающие особенности процессов распространения данных в ГИС корабля.

Структурно-функциональную модель ГИС корабля представим количество в виде множества параметров P , C и S :

$$G = f(P, C, S), \quad (1)$$

где G – структурно-функциональная модель ГИС корабля;

P – множество параметров, которые задаются как технические требования на составные элементы ГИС;

C – множество стоимостных характеристик составных элементов ГИС;

S – множество пространственных параметров (форм-факторов) составных элементов ГИС.

Тогда выбор структурно-функциональной модели ГИС корабля может быть сформулирована, как целочисленная задача условной многопараметрической оптимизации с ограничениями второго рода:

$$C \rightarrow \min_P \quad (2)$$

где C – стоимость ГИС корабля, которая является аддитивной функцией стоимостных характеристик ее составных элементов.

$$\bar{t}_{\text{дост}}(G) \leq T_{\text{доп}}, \quad (2)$$

где $\bar{t}_{\text{дост}}(G)$ – среднее время доставки в ГИС корабля;

$T_{\text{доп}}$ – ограничения на время доставки, рекомендуемые стандартами распространения пространственных данных.

$$S \leq S_{\text{доп}}, \quad (3)$$

где $S_{\text{доп}}$ – ограничения на выделяемые площади под ГИС корабля.

Набор параметров P характеризует определенное решение, принятое для построения комплекса технических средств ГИС корабля.

Между параметрами и временем доставки существует функциональная зависимость. Изменение значений P влечет изменение значения $\bar{t}_{\text{дост}}$. Таким образом, может быть решена обратная задача моделирования – определение или выбор из известного ряда значений множества P , таких, чтобы время доставки соответствовало рекомендованному.

Исходя из сформулированной задачи, определим дополнительные требования к исходным данным и результатам задачи:

1. Набор параметров должен максимально полно характеризовать степень выполнения основной функции ГИС корабля.

2. Для решения оптимизационной задачи выбора структурно-функциональной модели ГИС корабля:

– значения множеств P , C , S и значение $\bar{t}_{\text{дост}}$ должны иметь численные выражения.

– параметры и время доставки должны поддаваться аналитическому (возможно приближенному) расчету и практическим измерениям на имитационных моделях и реальных системах.

Выводы по первой главе

ГИС корабля представлена иерархией слоев: аппаратного, программного и информационного обеспечения. Каждый слой рассмотрен на уровне классификации элементов, образующих соответствующий слой ГИС и их характеристик:

– информационный слой – классификацией электронно-картографических систем;

– программный слой – классификацией специализированных и базовых информационных технологий, реализующих процессы обработки, передачи и хранения информации в ГИС.

– аппаратный слой – классификацией функциональных модулей, на которых построена ГИС корабля.

Предложена концептуальная модель ГИС корабля, представляющая собой интеграцию разнородных элементов на основе ЛВС с коммутацией сегментов.

Сформулирована задача диссертации, как задача разработки инженерной методики проектирования ГИС корабля под заданные цели плавания и учетом ограничений на требуемые показатели производительности ГИС при работе с актуальными данными и занимаемые площади.

Предложено решать задачу диссертации в два этапа:

1) средствами моделирования оценить время доставки данных, для чего надо предложить соответствующие модели и/или расчетные методики;

2) решить целочисленную задачу условной многопараметрической оптимизации выбора структурно-функциональной модели ГИС корабля.

2. Система моделей оценки передачи данных в ГИС корабля

Согласно приведенному в первой главе описанию концептуальной модели ГИС корабля время доставки данных по локальной сети, соединяющей элементы ГИС, представляет собой сумму:

$$t_{\text{дост}} = t_{\text{у.с}} + t_{\text{п.д}} + t_{\text{обр}},$$

где $t_{\text{у.с}}$ – время, необходимое на установление соединения;

$t_{\text{п.д}}$ – время передачи данных получателю;

$t_{\text{обр}}$ – время решения задачи.

Для оценки $t_{\text{у.с}}$ предлагается вероятностная модель установления соединения в ГИС корабля.

Оценку $t_{\text{п.д}}$ предлагается выполнить с применением аппарата систем массового обслуживания.

Оценку $t_{\text{обр}}$ предлагается выполнить с применением аппарата сети систем массового обслуживания.

2.1 Оценка времени установления соединения

Процесс установления соединения представляет собой посылку и прохождение вызова от источника к адресату по одному из множества альтернативных маршрутов. Каждый такой маршрут – это виртуальный канал (ВК), построенный от источника к адресату для последующей по нему передачи данных.

ВК состоит из транзитов (каналов), связывающих узлы коммутации (коммутаторы), входящих в данный маршрут. С точки зрения прохождения вызова от станции к станции состояние транзита бинарно: либо транзит занят, и вызов по нему не проходит (состояние транзита «1»), либо транзит имеет

канал требуемой скорости, и вызов через такой транзит пройдет (состояние транзита «0»).[8,9,16]

2.1.1 Математическая модель установления соединения в ГИС корабля

Конечный результат установления соединения имеет один из следующих вариантов:

- вызов дойдет до адресата и будет установлено соединение за время, не превышающее допустимое;
- вызов не дойдет до адресата, поскольку все альтернативные пути загружены или неработоспособные;
- дойдет до адресата и будет установлено соединение, но за время, превышающее допустимое.

В силу возможности рестартов, то есть возвращения вызова на предыдущие узлы – число транзитов, пройденных вызовом при его доставке к адресату, оказывается числом случайным. Данный аспект придает стохастичность процессу установления соединения, поэтому факт доставки вызова будем устанавливать с помощью имитации процесса доставки. [17,18]

Время установления соединения $t_{y.c}$ является случайной величиной, которая определяется выражением

$$t_{y.c} = \sum_{i=1}^{n_{tr}} t_{tr_i} + \sum_{i=1}^{n_{o.v}} t_{o.v_i} + n_p t_p, \quad (2.1)$$

где n_{tr} – число транзитов ВК, пройденных вызовом при установлении соединения;

$n_{o.v}$ – число транзитов, на которые пришлось вернуться вызову в режиме «обратной волны» для поиска альтернативного маршрута;

n_p – число рестартов в зафиксированной реализации процесса установления соединения; общем случае $0 \leq n_p < n_{\text{доп}}$, где $n_{\text{доп}}$ – допустимое число рестартов при установлении соединения;

$t_{\text{три}i}$ – время прохождения i -го транзита;

$t_{\text{о.в}i}$ – время прохождения i -го транзита в режиме «обратной волны»;

t_p – время рестарта.

При имитации доставки вызова каждый эксперимент на модели дает реализацию случайных величин $n_{\text{тр}}$, $n_{\text{о.в}}$ и n_p и в соответствии с выражением (2.1) оценивается время $t_{\text{у.с}}$, которое собственно и определяет результат установления (неустановления) соединения:

- если $t_{\text{у.с}} \leq t_{\text{доп}}$, то соединение установлено;
- если $t_{\text{у.с}} > t_{\text{доп}}$, то соединение установлено, но за время превышающее допустимое, и для данных срочной доставки это соответствует не установлению соединения, т.к. эти данные потеряли свою актуальность;
- если $n_p > n_{\text{доп}}$, то соединение не установлено.

Исходными данными для моделирования являются

1) структурная и потоковая метрики:

– множество альтернативных маршрутов (ВК), с описанием транзитов, составляющих каждый ВК: время передачи вызова по транзиту в режиме «прямой волны» и в режиме «обратной волны»;

– значения вероятностей полной занятости транзитов;

2) ограничения, при которых должно выполняться соединение:

- допустимое время установления соединения;
- число рестартов (попыток установления соединения);
- время выполнения рестарта.

Выходными данными оценки выполнения системных требований являются статистики, позволяющие оценить вероятность установления соединения за время, не превышающее допустимое, а также средние и среднеквадран-

тические значения $t_{y,c}$, характеризующие процесс установления соединения, которые могут быть использованы при упрощенных аналитических расчетах.

В качестве критерия эффективности этапа установления соединения прием вероятностное установление соединения за время, не превышающее допустимое. [10,39]

На всем множестве виртуальных каналов, соединяющих источник i и адресат j производится расслоение по числу занятых транзитов c ($c=c_{\min}, \dots, c_{\max}$). В отдельном эксперименте разыгрывают число c номеров занятых транзитов во множестве $ВК_{ij}$ и на полученной реализации $ВК_{ij}$ имитируется процесс доставки вызова адресату. По факту доставки вызова фиксируются значения $n_{тр}$, $n_{о.в}$ и $n_{р}$. Процедура повторяется N раз. По результатам экспериментов вычисляются необходимые статистики.

Реализация количества занятых транзитов c сводится к «выбору наугад» номеров транзитов из d возможных. Очередной номер занятого транзита z определяется по формуле $z = \lceil Ud + 1 \rceil$, где U – случайное число, $U \in [0,1]$, получаемое путем обращения к датчику случайных чисел. Скобки $\lceil \rceil$ означают округление в меньшую сторону. Транзиту с номером z присваивается «1» во множестве альтернативных маршрутов $ВК_{ij}$. Процедуру определения z повторяем c раз. [58,66,68]

Задача расчета характеристик установления (неустановления) соединения может быть сведена к задаче оценивания математического ожидания $M\xi$ случайной величины (СВ) $\xi=f(\alpha)$, причем СВ $\alpha=(\alpha_1, \dots, \alpha_d)$ имеет закон распределения вероятностей p (то есть $\alpha \sim p$), который известен. В данной задаче $\xi \in \{0,1\}$, $\xi=0$ соответствует установлению соединения, $\xi=1$ – неустановлению соединения. $M\xi=P\{\xi=1\}$ имеет смысл вероятности неустановления соединения. Случайная величина ξ невырожденная, то есть $0 < M\xi < 1$.

Случайную величину α интерпретируем как вектор $\alpha=(\alpha_1^v, \dots, \alpha_d^v)$, отображающий состояние транзитов, входящих в моделируемое множество альтернативных маршрутов с точки зрения наличия или отсутствия свободных

транзитов. Случайная величина $\alpha_i \in \{0,1\}$, $i = \overline{1,d}$, d – число транзитов в моделируемом множестве $ВК_{ij}$. Значение $\alpha_i=1$ имеет смысл отсутствия свободных каналов со скоростью υ в i -м транзите. Случайные величины α_i считаются независимыми. Случайная величина α имеет конечное множество значений $\alpha \in X$, $X = \{x_j; j = \overline{1,n}; n = 2^d\}$. Для нее распределение $p(x)$, $x \in X$ задается набором вероятностей $p(x_j) = P\{\alpha = x_j\} = p_j \geq 0, j = \overline{1,n}$. С учетом расслоения по c оценку $M\xi$ находим в виде:

$$\tilde{M}\xi = \sum_{c_{\min}}^{c_{\max}} M\xi(c), \quad (2.2)$$

где $\tilde{M}\xi$ – оценка вероятности не установления соединения при наличии c транзитов, непроводящих вызов.

Для задания состояний $ВК_{ij}(c)$ на k -м розыгрыше применяем случайный выбор номеров непроводящих транзитов c . В результате вектор α получает конкретную реализацию x , содержащую c единиц и $(d-c)$ нулей. В соответствии с правилом прохождения вызова по множеству альтернативных путей $ВК_{ij}$ и правилом установления соединения вычисляем значение

$$\xi(x|c)_k = f[ВК_{ij}^k],$$

где $\xi(x|c)_k$ – исход доставки (недоставки) вызова при k -м испытании (k -й реализации $ВК_{ij}$), $\xi(x|c)_k \in (0,1)$,

и вероятность полученной реализации

$$p(x) = p[ВК_{ij}^{(k)}].$$

Значение $\xi(x|c)_k=1$ имеет место, если вызов не дошел до адресата (все ВК из множества альтернативных маршрутов оказались непроводящими), либо число рестартов превысило допустимое значение, либо вызов дошел до адресата и установлено соединение, но за время $t_{y.c} > t_{\text{доп}}$.

Оценка вероятности неустановления соединения между парой станций (i,j) при c непроводящих транзитов во множестве всех ВК $_{ij}$ определяется в виде:

$$\tilde{M}_{\xi}(c) = \frac{C_d^c}{N_{2c}} \sum_{k=1}^{N_{2c}} p\left(\text{ВК}_{ij}^{(k)} \mid \xi(x|c) = 1\right),$$

где N_c – число разыгранных состояний (реализаций) множества ВК $_{ij}$.

Вероятность $p(\text{ВК}_{ij})$ рассчитывается по формуле:

$$p\left(\text{ВК}_{ij}^{(k)}\right) = \prod_{i=1}^h p(\alpha_i = 1) \prod_{\substack{j=1 \\ j \neq i}}^{d-h} [1 - p(\alpha_i = 1)].$$

Вероятности $p(\alpha_i=1)$, $i = \overline{1, d}$ заданы при решении задачи распределения потоков.

Окончательная оценка вероятности неустановления соединения между парой станций (i,j) имеет вид:

$$\tilde{M}_{\xi} = \sum_{c=c_{\min}}^{c_{\max}} \frac{C_d^c}{N_c} \sum_{k=1}^{N_c} p\left(\text{ВК}_{ij}^{(k)} \mid \xi(x|c) = 1\right). \quad (2.3)$$

Таким образом, выражение (2.3) полностью отражает вероятностный подход к определению времени установления соединения.

2.1.2 Эксперимент на модели

Пусть локальная сеть ГИС корабля задана следующей топологической структурой, приведенной на рисунке 2.1.

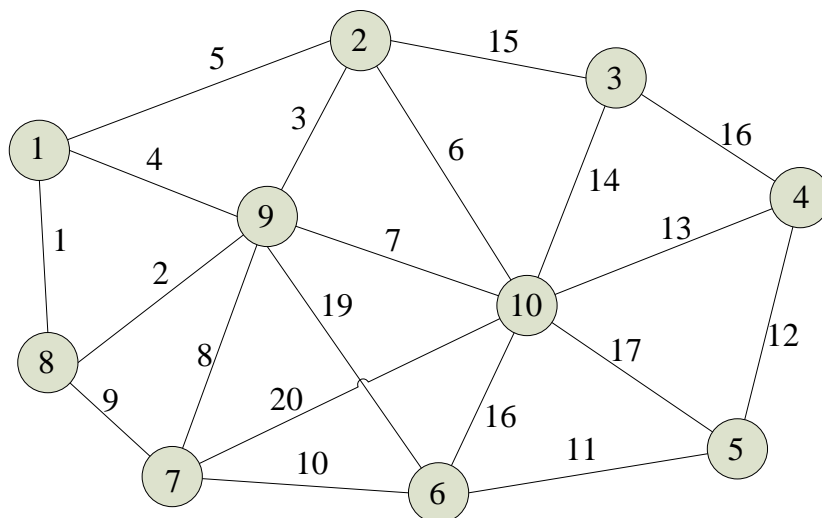


Рисунок 2.1 – Топология локальной сети ГИС корабля

Заданы следующие параметры, характеризующие коммутаторы и каналы связи:

- вероятность отказа коммутатора – 10^{-3}
- пропускная способность канала связи – 9600 бод
- вероятность отказа канала связи – $5 \cdot 10^{-3}$
- время прохождения i -го транзита $t_{\text{Тр}i} = 0,5$ мс;
- время прохождения i -го транзита в режиме «обратной волны» $t_{\text{о.В}i} = 0,7$ мс
- время рестарта $t_p = 10^{-2}$ мс
- допустимое время на установление соединения $t_{\text{доп}} = 50$ мс;
- количество экспериментов – 100.

Построим зависимости $\bar{t}_{\text{у.с}}$ от

- количества закрытых транзитов s ;
- количества рестартов n_p .

Собранные в результате моделирования статистики позволяют построить важные зависимости, характеризующие вероятность установления соединения в ГИС корабля.

На рисунках 2.2 – 2.4 приведены зависимости $t_{y.c}$ от числа неработоспособных каналов и коммутаторов в локальной сети ГИС и количества попыток на установление соединения.

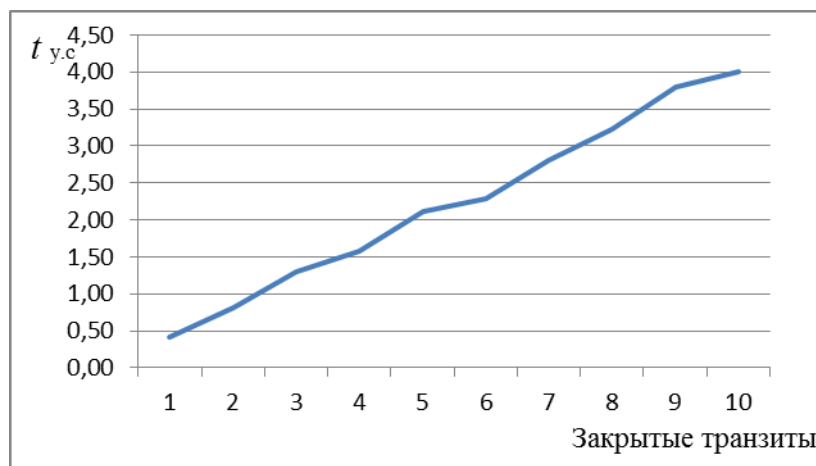


Рисунок 2.2 – Зависимость $t_{y.c}$ от числа закрытых транзитов

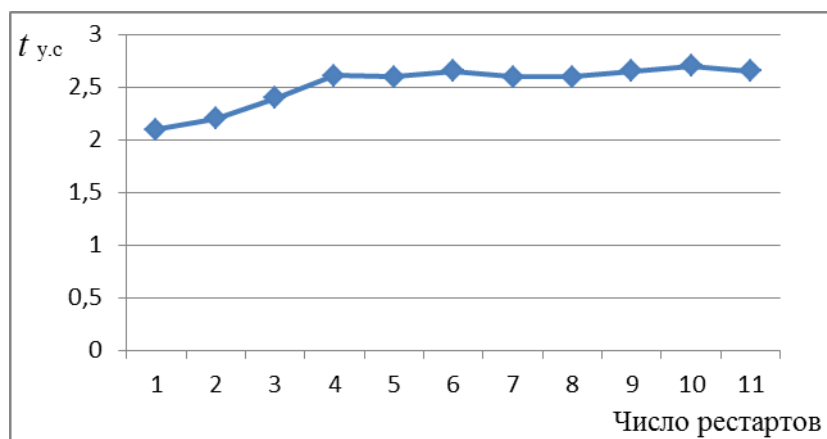


Рисунок 2.3 – Зависимость $t_{y.c}$ от числа рестартов

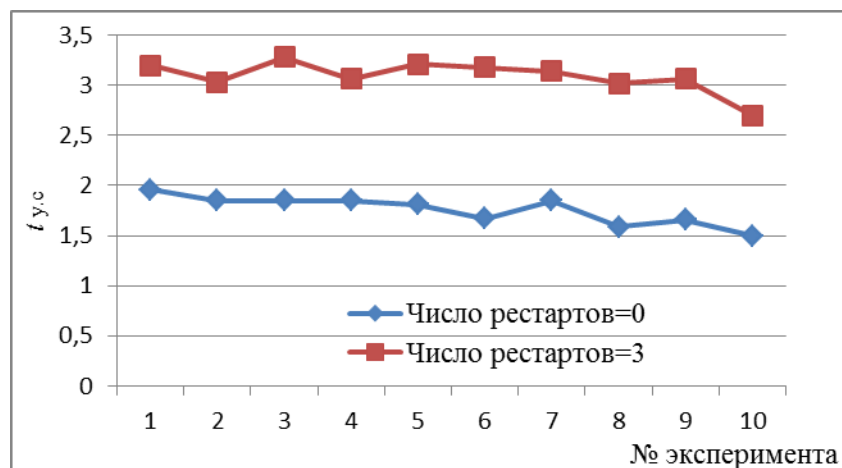


Рисунок 2.4 – Зависимость $t_{y,c}$ от числа рестартов

Выводы: с увеличением отказавших транзитов увеличивается время доставки пакета, т.к. происходит поиск альтернативных путей и соответственно тратится дополнительное время на переключение и на соединение между узлами. Вероятность доставки пакета при этом увеличивается, т.к. пакет, для которого выбран оптимальный маршрут, но в этом маршруте оказались вышедшие из строя коммутаторов или каналов связи будет отправлен по другому альтернативному маршруту.

То же самое можно сказать о влиянии количества рестартов: чем больше возможностей на повторный поиск альтернативных маршрутов (рестартов), тем больше времени пакет задерживается в сети. Однако при количестве отказавших транзитов, равное трем, увеличение количества рестартов уже не влияет на время доставки, т.к., видимо такое количество рестартов достаточно, чтобы найти альтернативный путь доставки пакета. Важно, что вероятность доставки пакета при этом увеличивается.

На рисунке 2.5 приведена зависимость количества установленных соединений от допустимого времени, отводимого на установление соединения при фиксированных $n_p=3$ и $c=3$, которая позволяет найти необходимые параметры, обеспечивающие гарантированное $t_{y,c}$.



Рисунок 2.5 – Зависимость количества установленных соединений от допустимого времени на установление соединения

Выводы: с увеличением времени, отводимого на доставку пакета, соответственно увеличивается количество установленных соединений, это означает, что вероятность доставки будет приближаться к единице. Таким образом, при определенных исходных данных можно находить гарантированное $t_{y.c.}$

Определим зависимость вероятности установления соединения от числа неработоспособных транзитов и коммутаторов. На рисунках 2.6 и 2.7 приведены соответствующие графики. Показания снимались при количестве рестартов $n_p=3$.

Как и следовало ожидать, вероятность будет снижаться с увеличением количества неработоспособных каналов (транзитов) и коммутаторов. В этом случае, для того чтобы пакеты все же дошли до места назначения необходимо из построенного множества виртуальных каналов назначать тот, который не содержит оборванных путей и неработоспособных коммутаторов. Для этого администратору надо изменить приоритет отправки. В имитационной модели такая возможность предусмотрена (рисунок 2.8).

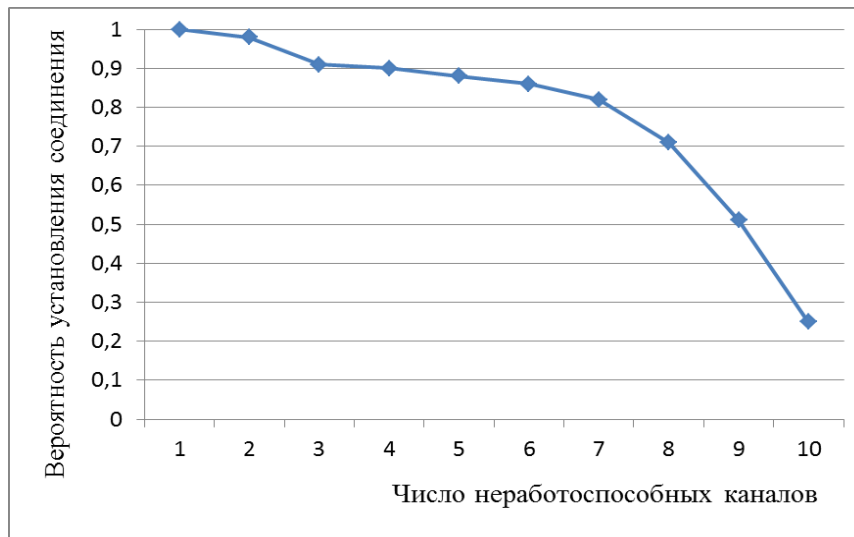


Рисунок 2.6 – Зависимость вероятности установления соединения от числа неработоспособных транзитов

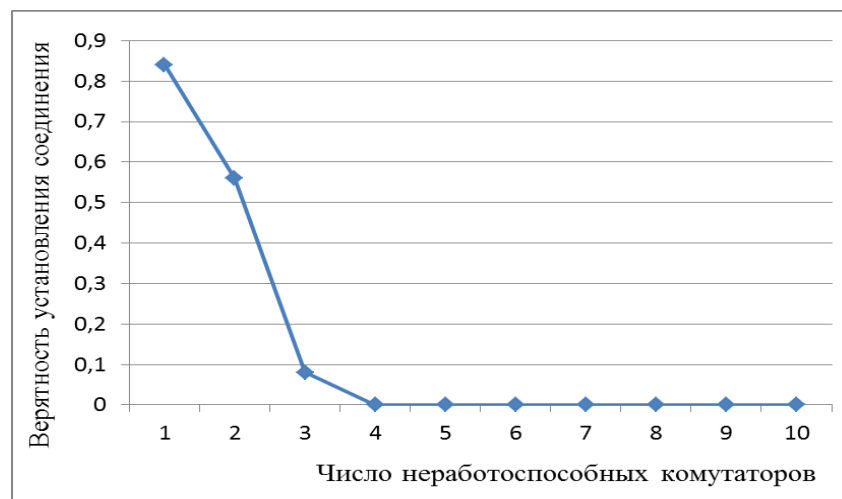


Рисунок 2.7 – Зависимость вероятности установления соединения от числа неработоспособных коммутаторов

Выводы: с увеличением числа неработоспособных транзитов вероятность установления соединения будет снижаться. Это означает, что надо либо увеличивать количество рестартов, либо назначать другие приоритеты отправки. С увеличением количества неработоспособных коммутаторов сеть перестает выполнять свою основную функцию, и вероятность установления соеди-

нения быстро сходит к нулю. Данный эксперимент позволяет определить максимальное количество неработоспособных коммутаторов, при котором сеть все еще будет функционировать, т.к. еще будут существовать альтернативные маршруты.

На рисунке 2.8 приведен пример того, что для разных приоритетов отправки вызова на установление соединения время установления соединения будет разное. Естественно, что всегда выстраивается оптимальный маршрут, но если он не работает, значит можно назначить другой маршрут, возможно худший по времени, но с гарантией установления соединения.

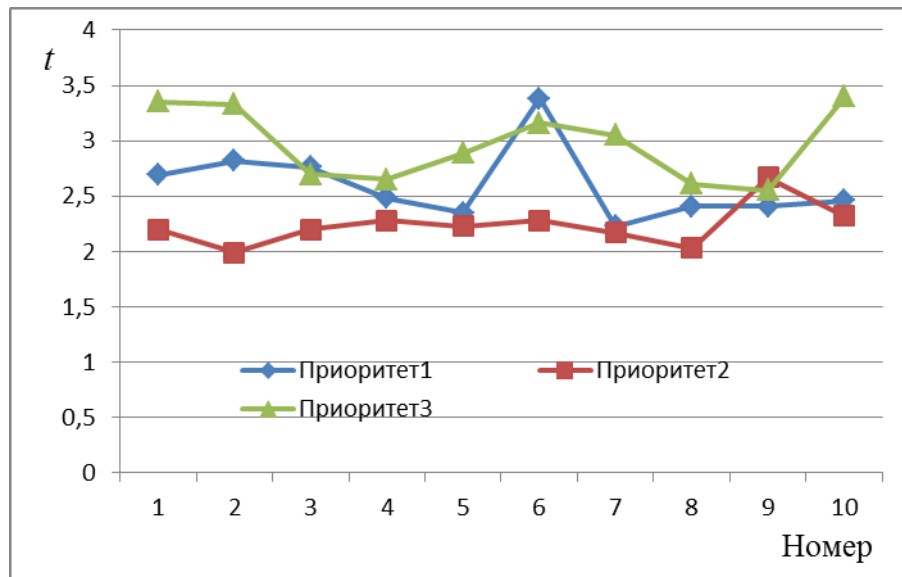


Рисунок 2.8 – Зависимость $t_{y.c}$ от приоритета отправки

В имитационной модели также предусмотрена возможность изменять допустимое время установления соединения, и по результатам эксперимента рекомендовать его значение. Например, построим зависимости вероятности и условной вероятности установления соединения от числа неработоспособных транзитов (рисунок 2.9).

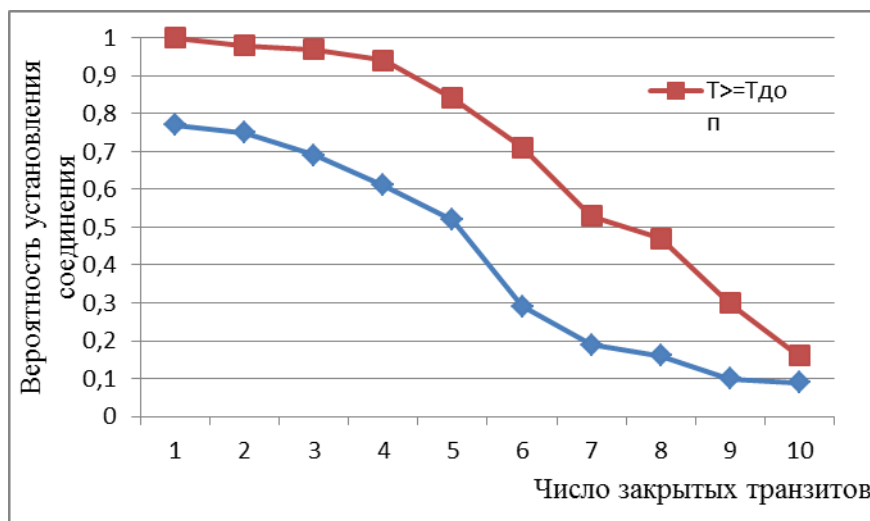


Рисунок 2.9 – Зависимость вероятности и условной вероятности установления соединения от числа неработоспособных транзитов

Вывод: Для срочных данных имеет значение допустимое время установления соединения доставки. Графики демонстрируют, что даже если соединение будет установлено после нескольких попыток, то есть вероятность, что сообщение придет позже назначенного времени. Для некоторых видов трафика такая характеристика имеет большое значение, и имитационная модель позволяет оценить такую характеристику, как условная вероятность – вероятность установления соединения при условии, что время отводимое для этого меньше допустимого.

2.2 Оценка времени передачи данных

Оценка времени передачи данных в ГИС корабля $t_{п,д}$ необходима для определения варианта комплексирования клиентской части. Для оценки $t_{п,д}$ воспользуемся аппаратом теории очередей.

2.2.1 Математическое обеспечение расчета времени передачи данных

Математической базой оценки временных характеристик распределенных систем, какой собственно и является ГИС корабля в, частности являются системы массового обслуживания – СМО.

Аппарат СМО позволяет получить значения временных характеристик в аналитическом виде. Наиболее разработана теория экспоненциальных СМО, и известны практические формы расчета временных характеристик экспоненциальных СМО.

Важнейшая характеристика СМО – вероятности их состояний. Для определения вероятностей состояний исследуют протекающий в СМО случайный процесс. В качестве моделей протекающих в СМО процессов наиболее часто используют марковские и полумарковские процессы.

Марковским процессом с непрерывным временем описывают функционирование экспоненциальных СМО. СМО называется экспоненциальной, если входящие потоки требований пуассоновские, а время обслуживания имеет экспоненциальное распределение. [74,75]

Однако особенности применения теории СМО к моделированию сетей вызывают необходимость учитывать в этой теории различные виды взаимодействия отдельных СМО в сети. Взаимодействие СМО является следствием того, что каждое требование подлежит обслуживанию на нескольких узлах. Основной вид взаимодействия узлов - последовательное поэтапное обслуживание требования в разных узлах. Другие виды взаимодействия могут заключаться в зависимости от длительностей пребывания на последовательных этапах, в одновременном занятии требованием на некотором этапе двух или нескольких узлов сети, в блокировках узла и т.д.

СМО отображают функционально самостоятельные части ГИС корабля, связи между СМО – структуру локальной сети ГИС, а требования СМО – разнородные данные, циркулирующие в ГИС и подробно рассмотренные в разд. 1.1.

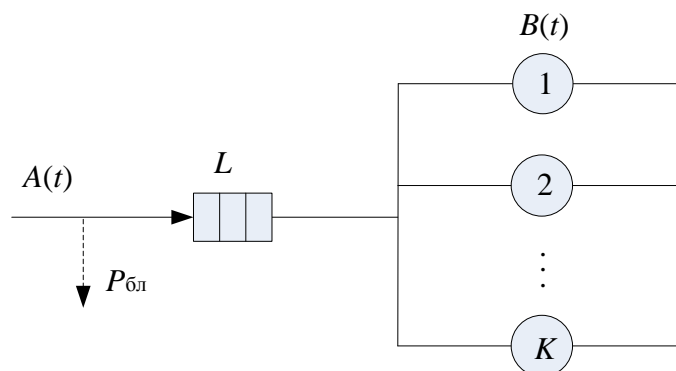


Рисунок 2.10 – Система массового обслуживания

СМО задается следующими параметрами:

$A(t)$ – функцией распределения длительности интервала между поступлениями заявок, $0 \leq t$;

$B(t)$ – функцией распределения длительности обслуживания заявок;

K – количеством приборов (каналов) обслуживания, $K \geq 1$;

L – число мест в очереди, $0 \leq L \leq \infty$.

Показатели производительности, которые позволяет оценить аппарат СМО:

$P_{\text{бл}}$ – вероятность блокировки (потери) заявки;

$\bar{T}_{\text{ож}}$ – среднее время ожидания начала обслуживания;

\bar{L} – средняя длина очереди;

\bar{M} – среднее число заявок в СМО.

В теории СМО для оценки производительности применяется аппарат преобразования Лапласа-Стилтьеса (ПЛС). []

Преобразованием Лапласа-Стилтьеса функции распределения $B(t)$ называется функция $\beta(s)$, определяемую следующим образом[58]

$$\beta(s) = \int_0^{\infty} e^{-st} dB(t), \quad (2.3)$$

где s – параметр ПЛС.

Отметим некоторые свойства ПЛС, необходимы при оценке времени передачи данных:

1. Важным свойством преобразования ПЛС является то, что преобразование Лапласа-Стилтьеса суммы случайных величин равно произведению преобразований Лапласа-Стилтьеса каждой из этих величин, то есть если две независимые случайные величины имеют ПЛС $\beta_1(s)$ и $\beta_2(s)$ их функций распределений, то ПЛС функции распределения суммы этих величин является $\beta_1(s)\beta_2(s)$. [58]

2. Если B_k есть k -й момент случайной величины относительно начала координат, то

$$B_k = (-1)^k \left. \frac{d^k \beta(s)}{ds^k} \right|_{s=0} \quad (2.4)$$

то есть моменты случайной величины определяются дифференцированием в нуле (при $s=0$) соответствующее число раз преобразования Лапласа-Стилтьеса функции распределения этой величины. Первый центральный момент определяет математическое ожидание случайной величины, [58]

$$\bar{t} = \beta = - \left. \frac{d\beta(s)}{ds} \right|_{s=0}, \quad (2.5)$$

а второй момент нужен для нахождения дисперсии случайной величины []

$$\sigma^2 = B_2 - B_1^2 = \left. \frac{d^2 \beta(s)}{ds^2} \right|_{s=0} - \left[\left. \frac{d\beta(s)}{ds} \right|_{s=0} \right]^2. \quad (2.6)$$

$$3. \lim_{s \rightarrow 0} \beta(s) = \lim_{t \rightarrow \infty} B(t).$$

4. Вероятностный смысл преобразования Лапласа-Стилтьеса. Величина

$e^{-st} B(t_i)$ есть вероятность сложного события, состоящего в том, что случайная величина не превысит значения t_i (сомножитель $B(t_i)$), а кроме того, за время t_i не произойдет ни одной «катастрофы» (сомножитель e^{-st}). Параметр s рассматривается как интенсивность «катастроф». Интегрирование по всему диапазону дает $\int_0^{\infty} e^{-st} dB(t) = \beta(s)$. Таким образом, вероятностный смысл преобразования

Лапласа-Стилтьеса состоит в том, что оно определяет вероятность того, что за время t_i не произойдет ни одной «катастрофы». [58]

Распределение времени передачи данных в локальной сети ГИС корабля описывается в терминах преобразования Лапласа-Стилтьеса (ПЛС). В ГИС одновременно передаются между различными ее элементами K независимых пуассоновских потоков данных различных классов интенсивности λ_k , $k=1, \dots, K$. Длительности передачи сообщений каждого класса – независимые случайные величины с распределениями

$$B_k(t) = 1 - e^{-\mu_k t}, \quad (2.7)$$

где $\mu_k = C / L_k$ – пропускная способность канала в пакетах данных,

C – пропускная способность канала в битах,

L_k – длина пакета данных k -го класса, $k=1, \dots, K$.

Определим ПЛС длительности передачи сообщения k -го класса от абонента i к абоненту j .

Пусть $\pi_{ij} = \{e_d, d=1, \overline{N}\}$ – множество элементов маршрута ij , N – мощность этого множества, l – номер элемента маршрута, вносящего наибольшую задержку, то есть «узкое» место маршрута. [58]

На рисунке 2.11 показан вероятностный граф, представляющий собой отображение всех возможных вариантов прохождения заявки от источника к получателю.

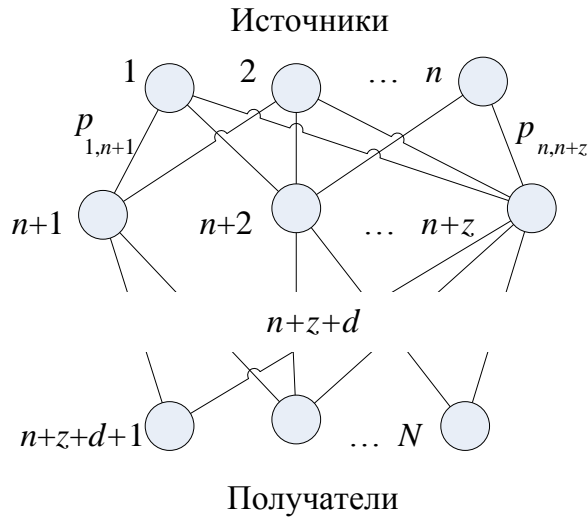


Рисунок 2.11 – Вероятностный граф перехода заявки

Тогда соответствующее ПЛС будет иметь вид

$$\beta_k(t) = \beta_{kl}(t) \prod_{\substack{d=1 \\ d \neq l}}^N \beta_d(t), \quad (2.8)$$

где $\beta_{kl}(t)$ – ПЛС длительности пребывания сообщения k -го класса в «узком» месте маршрута,

$\beta_d(t)$ – ПЛС длительности пребывания пакета в d -м элементе маршрута.[]

Данное распределение будет функцией от векторов интенсивностей поступления и обслуживания сообщений на элементы маршрута $\beta_k(t) = f(\lambda_k, \mu_k)$.

Таким образом, предложенная модель дает распределение времени пребывания сообщения k -го класса в локальной сети корабля, построенной на основе технологии коммутации сегментов и узлов, при заданном векторе вероятностей возникновения ошибок на элементах маршрута для всех пар сетевых абонентов.[58,66]

2.2.3 Эксперимент по оценке времени передачи данных

В разд. 2.2.1 с применением характеристического преобразования Лапласа-Стилтьеса может быть определено «узкое» место маршрута от источника i к приемнику j .

ПЛС позволяет при ограничениях на время $t_{п.д}$ некоторого маршрута определить возможную нагрузку на этот маршрут и осуществить выбор соответствующего оборудования.

На рисунке 2.14. приведены результаты ПЛС для $t_{п.д}$ при а) $\mu=10$ пакетов/с и б) $\mu=0,5$ пакетов/с.

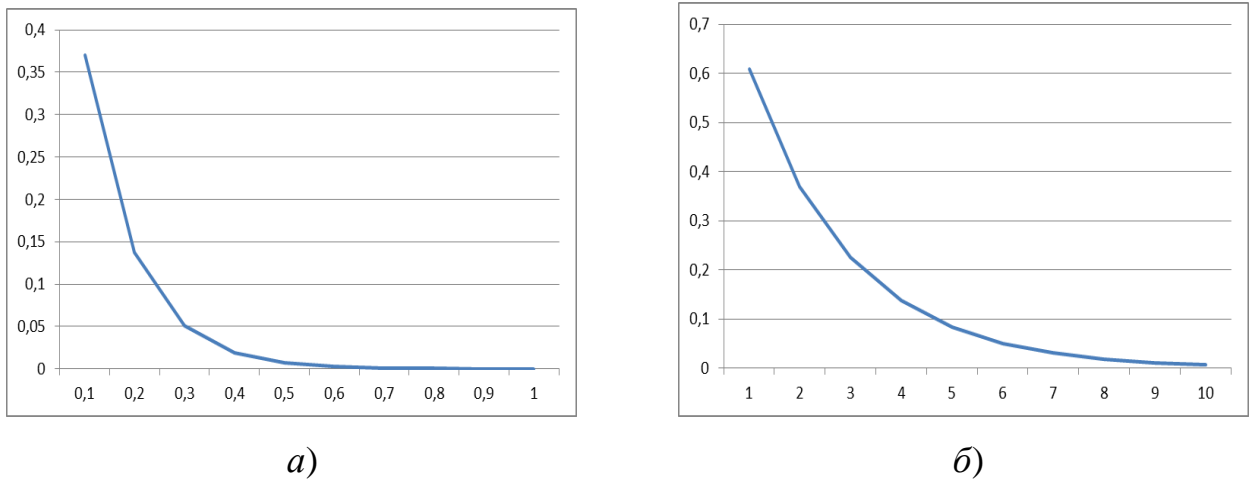


Рисунок 2.14– Распределение ПЛС

а) $\mu=10$ пакетов/с и б) $\mu=0,5$ пакетов/с

При пропускной способности $\mu=10$ пакетов/с и допустимом уровне «катастроф», например занятости узла, сбой программы и т.п. время выполнения составит более 0,3 с. При пропускной способности $\mu_k=0,5$ пакета/с время выполнения k -го этапа маршрута при тех же условиях составит более 7 с.

2.3 Оценка времени решения задачи

Для оценки времени решения задачи в ГИС корабля, то есть составляющую $t_{\text{обр}}$ воспользуемся аппаратом сети систем массового обслуживания.

2.3.1 Математическое обеспечение расчета времени решения задачи

Математической базой расчета временных характеристик решения задачи в АРМ, являющимся конечным устройством ГИС корабля, будет разомкнутая сеть массового обслуживания – РСМО.

Сеть массового обслуживания представляет собой совокупность конечного числа N обслуживающих узлов, в которой циркулируют заявки, переходящие в соответствии с маршрутной матрицей из одного узла в другой. Переход заявок между узлами происходит мгновенно в соответствии с переходными вероятностями $p_{ij}, i, j = \overline{1, N}$, p_{ij} – вероятность того, что заявка после обслуживания в узле i перейдет в узел j .

Разомкнутая сеть является сеть, в которую заявки поступают из внешней среды и уходят после обслуживания из сети во внешнюю среду. Другими словами, особенностью разомкнутой СМО (РСМО) является наличие одного или нескольких независимых внешних источников, которые генерируют заявки, поступающие в сеть, независимо от того, сколько заявок уже находится в сети. В любой момент времени в РСМО может находиться произвольное число заявок (от 0 до ∞). [32,48]

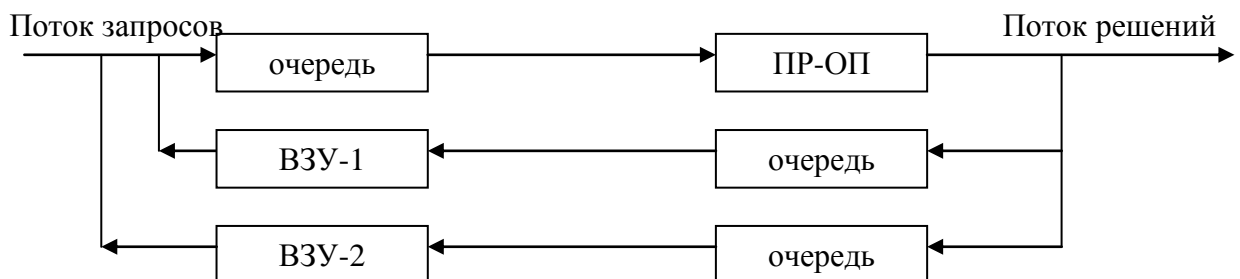


Рисунок 2.15 – Структура АРМ

ПР-ОП – процессор – оперативная память (системный блок);

ВЗУ-1 – хранит данные;

ВЗУ-2 – хранит программы задач.

Среднее время решения задания в СеМО рассчитывается по формуле:

$$t_{\text{обр}} = \frac{1}{I} \sum_{j=1}^N \lambda_j T_j^{\text{преб}} \quad (2.10)$$

где I – интенсивности входных потоков

$T_j^{\text{преб}}$ – время пребывания отдельного обращения на отдельном устройстве

λ_j – интенсивность потока заявок

N – Число СМО

Для определения интенсивности $\lambda_1, \lambda_2, \lambda_3$ и время пребывания заявки в СМО $T_1^{\text{пр}}, T_2^{\text{пр}}, T_3^{\text{пр}}$ необходимо составить уравнение баланса сети с учетом свойств слияния и ветвления потоков. С учетом структуры АРМ изображенной на рисунке 2.15 получаем матрицу P переходов:

$$P = \begin{matrix} & \begin{matrix} 0 & 1 & 2 & 3 \end{matrix} \\ \begin{matrix} 0 \\ P=1 \\ 2 \\ 3 \end{matrix} & \begin{matrix} \begin{matrix} 0 & 1 & 0 & 0 \\ P_{10} & 0 & P_{12} & P_{13} \\ 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{matrix} \end{matrix}$$

$$P_{10} = \frac{1}{D+2}; \quad P_{12} = \frac{\bar{D}}{D+2}; \quad P_{13} = \frac{1}{D+2}, \quad (2.11)$$

где D – число обращений к ВЗУ.

Определим время обслуживания отдельного запроса.

Время обслуживания отдельного обращения при быстройдействии процессора $w_i, i=1, \dots, I$ и трудоемкости обслуживания отдельного решения \bar{Q}_1

$$\bar{T}_{\text{обс}}(i) = \frac{\bar{Q}_1}{W_i} \quad (2.12)$$

Время пребывания отдельного обращения (требования) в соответствующих СМО

$$T_1^{\text{np}} = \frac{1}{\mu_1 (1 - \rho_1)} = \frac{\bar{T}_{\text{обсл}}^{(i)}}{1 - \rho_1^{(i)}} = \frac{\bar{Q}_1}{W_1 - \lambda_1 Q_1}; \quad (2.13)$$

$$T_2^{\text{np}} = \frac{1}{\mu_{\text{взyl}} (1 - \rho_2)} = \frac{1}{\mu_2 - \lambda_2}; \quad (2.14)$$

$$T_3^{\text{np}} = \frac{1}{\mu_{\text{взy2}} (1 - \rho_3)} = \frac{1}{\mu_3 - \lambda_3}. \quad (2.15)$$

Таким образом, можно определить передаточные коэффициенты a_i , $i = 1, 2, 3$ и времена пребывания отдельных требований $T_1^{\text{np}}, T_2^{\text{np}}, T_3^{\text{np}}$ в соответствующих СМО. Подставим выражения этих величин в (13) и определим время решения задачи на АРМ

$$t_{\text{обр}}^{(i)} = \sum_{j=1}^3 a_j T_{\text{гппр}}^{(i)} = (4 \times T_1^{\text{np}} + 2 \times T_2^{\text{np}} + T_3^{\text{np}}) = (\bar{D} + 2) \times \frac{\bar{Q}_1}{W_1 - \lambda_1 Q_1} + \bar{D} \times \frac{1}{\mu_2 - \lambda_2} + \frac{1}{\mu_3 - \lambda_3}, \quad (2.16)$$

2.2.2 Эксперимент на модели

Пусть АРМ как модуль ГИС корабля задан следующей формализованной схемой массового обслуживания:

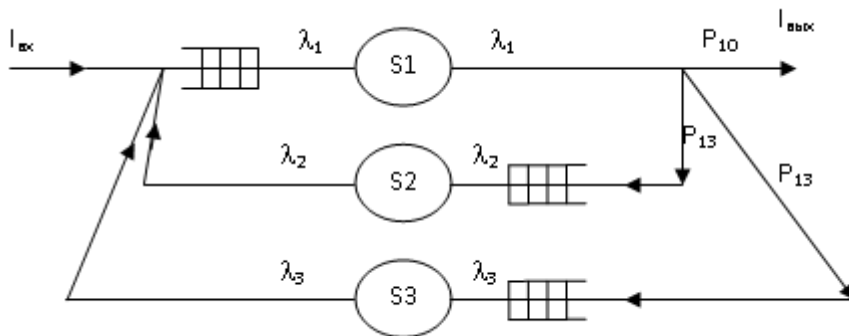


Рисунок 2.16 – Формализованная схема массового обслуживания АРМ

Заданы следующие параметры, характеризующие АРМ:

Множество процессоров w_i характеризующиеся быстродействием операции/с

Трудоёмкость обслуживания отдельного решения 65535 операций в секунду

Число обращений к ВЗУ 100

Интенсивность обслуживания заявок 0,0085, 0,0089, 0,0095 1/с

Найти устройство из ряд процессоров удовлетворяющих условию $t_{обр} < 2с$

Построим зависимости $t_{обр}$ от

- производительности процессора;
- скорости считывания ВЗУ.

На рисунке 2.17 показана зависимость времени ответа от частоты процессора.

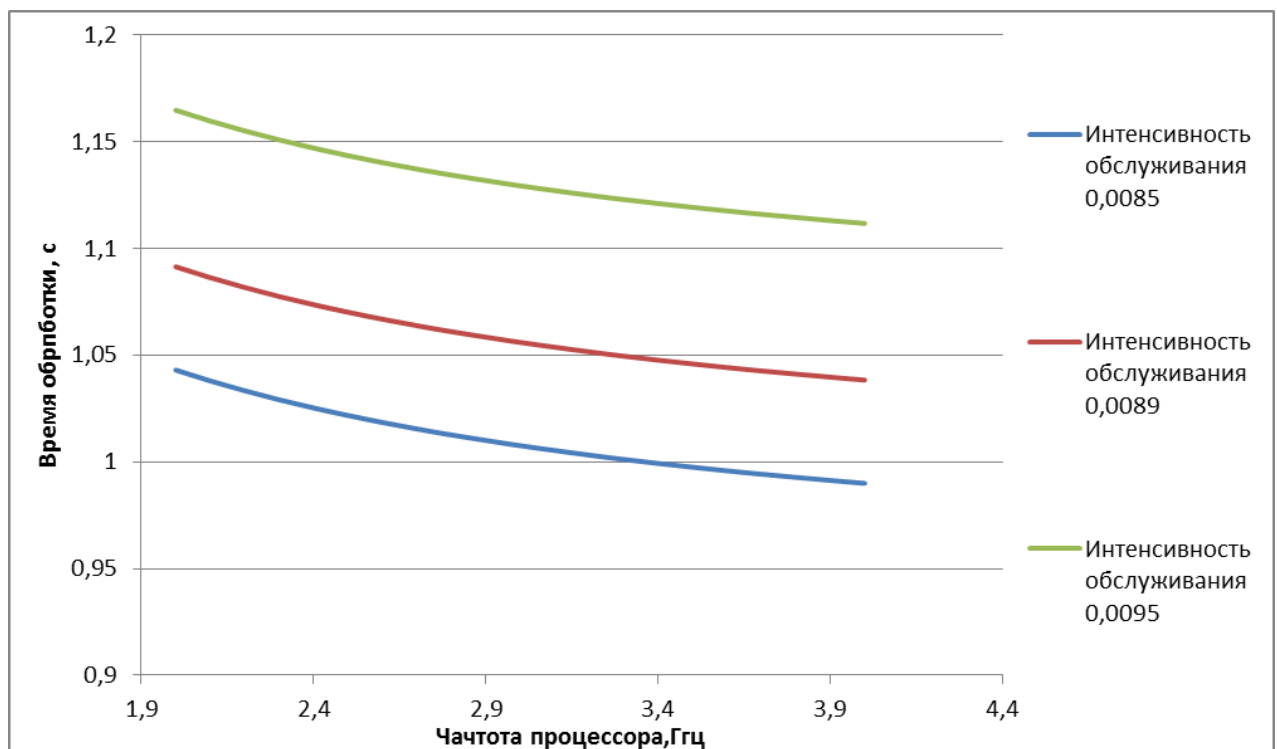


Рисунок 2.17 – Зависимость времени ответа от частоты процессора

Модель позволяет оценить зависимость времени обработки от других характеристик АРМ, что позволяет для директивного времени ответа выбрать нужные варианты процессор и памяти.

Выводы по второй главе

Предложена вероятностная модель установления/неустановления соединения между источником и адресатом в ГИС корабля, которая может быть инструментом, позволяющим решать задачу оценки надежной передачи данных. В модели учтены условия, соответствующие реальному процессу установления соединения – наличие неработоспособных каналов и узлов, ограниченное количество попыток установления соединений, наличие альтернативных маршрутов.

Эксперимент на модели установления/неустановления соединения между источником и адресатом в ГИС корабля показывает, что возможно решение как прямой задачи моделирования – определение времени установления соединения, так и обратной – подбор параметров для обеспечения допустимого времени установления соединения при соблюдении вероятности гарантированной доставки вызова.

Предложено для оценки времени передачи данных применить аппарат преобразования Лапласа-Стилтьеса (ПЛС). Первый центральный момент ПЛС позволяет определить среднее время передачи данных по маршруту сети ГИС корабля. Вероятностный смысл ПЛС позволяет выполнить выбор аппаратной составляющей построения сети ГИС корабля.

Предложено для оценки времени решения задачи применить аппарат сетей систем массового обслуживания. Модель позволяет определить время решения задачи на АРМ, учитывая интенсивность входного потока, вероятность решения задачи на каждом устройстве и временные характеристики.

3 Методика проектирования структурно-функциональной модели ГИС корабля

Согласно приведенному в первой главе описанию концептуальной модели ГИС корабля, наибольший эффект в реализации возможностей базовых информационных технологий и поддерживающих их технических и программных средств может быть получен на пути интеграции образцов РЭС, навигационного оборудования и электронных карт в единую геоинформационную систему. Таким образом, судоводитель будет иметь возможность визуально следить за поступающей информацией и оперативно реагировать на нее в районе следования.[14,21]

Учитывая требования частей «Радиооборудование» и «Навигационное оборудование», «Правил по оборудованию морских судов» РМРС и требования МК СОЛАС 74 с поправками, определение минимального состава радио и спутникового оборудования определяется:

- морским районом плавания;
- категорией корабля

Следовательно, исходными данными для формирования элементов интегрированной геоинформационной системы поддержки принятия решения управления кораблем будут служить назначение корабля, после чего необходимо будет выбрать один из морских районов следования, который доступен для выбранного типа корабля.

Зная минимальный набор РЭС для формирования элементов интегрированной геоинформационной системы, навигационного оборудования и можно судить о составе и характеристиках компонентов сети передачи данных и АРМ.

Приведённые во второй главе математическая модель установления соединения в ГИС корабля и математическое обеспечение расчета времени передачи данных позволяют определить характеристики компонентов сети передачи данных.

Определение допустимых вариантов комплексирования аппаратуры передачи данных и систем компьютерных блоков, для работы АРМ сводится к нахождению его компонентов, при ограничении $t_{\text{обр}} \leq T_{\text{доп}}$.

При невыполнении директивных требований обработки информации проводится оптимизация структурного состава ГИС с использованием генетического алгоритма описанного во второй главе.

Поскольку задачей диссертации является разработка инженерной методики проектирования ГИС корабля, опишем этапы и шаги, достаточные для получения структурно-функциональной модели ГИС корабля. Согласно этой необходимо определить исходные данные, временные характеристики и в зависимости от расчетов, либо сформировать полный списочный состав оборудования для построения структуры системы, либо определить узкое место системы

3.1 Определение исходных данных для построения структуры системы

Исходными данными будем считать: назначение объекта и морской район.

С точки зрения эксплуатации наиболее важным является деление судов по назначению, поскольку в последнее время быстро развивается специализация судов.

Они делятся на:

- транспортные (пассажирские, грузовые, грузопассажирские);
- промысловые (рыбодобывающие, рыбоперерабатывающие)
- научно-исследовательские (экспедиционные, гидрографические)
- учебные и спортивные, специальные (лоцманские, плавучие маяки, водолазные, пожарные)
- судоремонтные (плавучие мастерские, подъемные краны, доки)
- служебные (ледоколы, буксиры, толкачи, разъездные)
- спасательные (базы, боты, понтоны, буксиры)

- технические (грунтово­зы, дноуглубители)

Для данной работы выделим первые 3 вида назначения судов: транспортные, промысловые и научно-исследовательские.

Определение категории, позволяет определить количество АРМ и предполагаемый объем ГИ. Так, научно-исследовательскому судну потребуется минимум 3 автоматизированных рабочих места, 2 определяют работу на мостике и 1 – для научно-исследовательских работ

Минимальный состав радиооборудования на корабле дает морской район, который в Российской Федерации определяется Глобальной морской системой связи при бедствии (ГМССБ). Во всех районах ГМССБ должна быть обеспечена постоянная возможность аварийного оповещения. Для этих целей ИМО разработаны минимальные требования к составу радиооборудования в зависимости от района плавания и его размещению, а также эксплуатационные требования к этому оборудованию.

Поэтому после определения назначения корабля, определяем район плавания (рисунок 3.1)

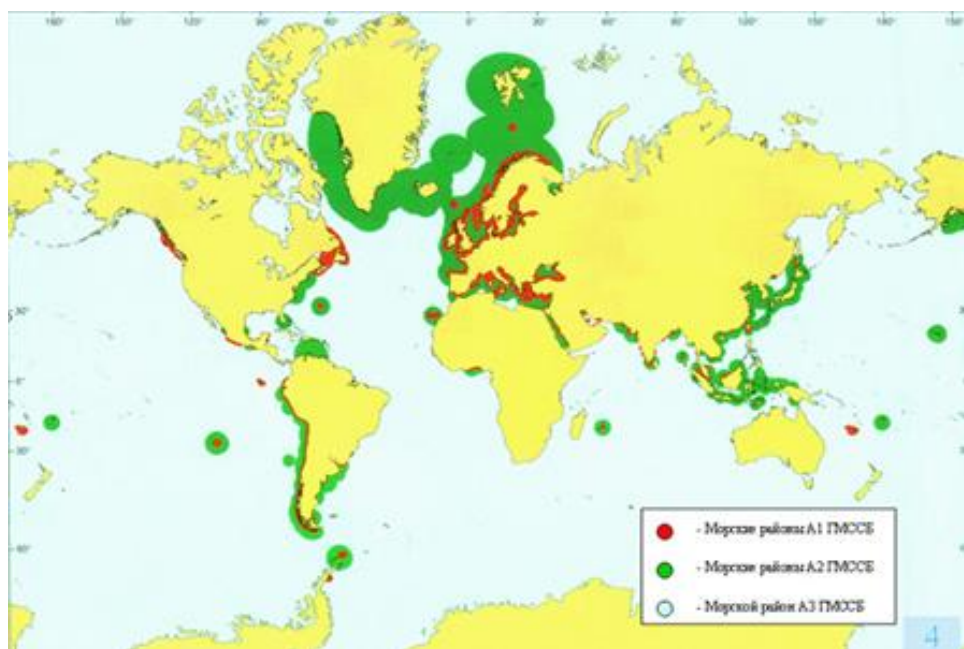


Рисунок 3.1 – Районы плавания

Этих районов всего четыре:

Морской район А1 – район в пределах зоны действия по крайней мере одной береговой УКВ радиостанции, обеспечивающей постоянную возмож-

ность оповещения о бедствии с использованием ЦИВ на 70-м канале (20–30 миль);

Морской район А2 – район, за исключением морского района А1, в пределах зоны действия по крайней мере одной береговой ПВ радиостанции, обеспечивающей постоянную возможность оповещения о бедствии с использованием ЦИВ (около 100 миль);

Морской район А3 – район, за исключением морских районов А1 и А2, в пределах зоны действия геостационарных спутников ИНМАРСАТ (примерно между 70° северной широты и 70° южной широты);

Морской район А4 – район, находящийся за пределами морских районов А1, А2 и А3.

Минимальный состав радиооборудования в зависимости от района плавания приведен на рисунке 3.2.

Район			оборудование
А1	А2	А3	УКВ радиостанция АИС Приемник ЦИВ Приемник НАВТЕКС Аварийный радиобуй
			А4 Носимые УКВ СЧ/ПВ – радиоустановка с радиотелефоном Судовая земная станция «Инмарсат-С» с приемником расширенного группового вызова
			ПВ/КВ – радиоустановка с радиотелефоном, ЦИВ и УБПЧ ПВ/КВ – радиоустановка для радиосообщений общего назначения
			Радиопередатчик 500 КГц Радиоприемник 500 КГц

Рисунок 3.2 – Минимальный состав радиооборудования в зависимости от района плавания

Так для океанографического исследовательского судна «Адмирал Владимирский» – построенного по проекту 852 на Щецинской верфи им. Барского в 1975 г. в Польше, имеющего следующие характеристики: размеры и параметры: длина составляет 147,8 м, ширина – 18,6 м, осадка – 6,4 м, водоизмещение – 9120 т, максимальная скорость – 19 узлов, дальность плавания – 18000 миль при 15 узлах, следующего из Санкт-Петербурга в пределах зоны А1, минимальный состав для выхода в море будет следующим:

- 3 АРМ
- УКВ радиостанция
- Автоматическая идентификационная система (АИС)
- Приемник ЦИВ
- Приемник НАВТЕКС
- Аварийный радиобуй



Рисунок 3.3 – Маршрут следования судна «Адмирал Владимирский»

Приемниками геоданных в таком случае служит АИС, наблюдения за течениями, химическая гидрология, морские биологические наблюдения, морская метеорология, актинометрические наблюдения, наблюдения за волнением, аэрологические наблюдения (M01).

3.2 Определение временных характеристик

Требования к времени обработки информации будут вытекать из времени затрачиваемом на прием пространственной информации с ФЭ1, ФЭ2 и ФЭ3 приведенных в первой главе в структурно-функциональной модели ГИС.

Процесс обнаружения объектов в пространстве и определение их координат радиотехническими методами называется радиолокацией. Приборы, обеспечивающие радиолокацию объектов в пространстве, называются радиолокационными станциями (РЛС).

Для радионавигации на море используется и совершенствуется лишь один вид – активная импульсная двухкоординатная РЛС.

Морские навигационные РЛС измеряют два параметра в полярной системе координат: расстояние до объекта и направление на объект (курсовой угол или пеленг).

Измерение расстояний производится амплитудным (импульсным) способом. Расстояние до объекта определяется измерением времени t_D от момента излучения «зондирующего» импульса до приема соответствующего отраженного импульса. Время t_D определяется как время прохождения импульса до объекта и обратно:

$$t_D = \frac{2 \cdot D}{c}, \quad (3.1)$$

где D – расстояние до объекта

c – скорость распространения радиоволн

Чем дальше объект следования отдалается от источника распространения сигнала, тем больше ему времени необходимо на прием сигнала по радиоканалу, поэтому скорость обработки информации на рабочей станции должна увеличиваться. Увеличение скорости обработки информации на рабочей станции позволит компенсировать затраты по времени на прием, и тем самым увеличит скорость принятия решения лицом управляющим кораблем. Из этого

можно сделать вывод, что допустимое время передачи данных будет прямо пропорционально удаленности источника распространения

Зная максимальную удаленность от источника распространения в морских районах плавания определим t_D для каждого:

В морском районе А1 максимальное расстояние от берегового источника составляет 30 миль, или приблизительно 48,28 км. Зная скорость распространения радиоволн $t_D=32,1*10^{-5}$ с.

В морском районе А2 максимальное расстояние от берегового источника составляет 100 миль, или приблизительно 160 км, следовательно $t_D=106*10^{-5}$ с.

В морском районе А3 и А4 для навигации используется спутниковые систем, в таком случае максимальное расстояние от источника составляет 20000 км, следовательно $t_D=0,65*10^{-1}$ с.

Учитывая проведенные эксперименты во второй главе, время установления соединения от приемника до конечного абонента (АРМ) не превышает 50 мс. Тогда полное время доставки сообщения до АРМ будет складываться из времени установления соединения и времени передачи данных до приемника, определим время доставки в каждом районе плавания:

для морского района А1 $t_{\text{дост}}$ составит $5,032*10^{-2}$ с

для морского района А2 $t_{\text{дост}}$ составит $5,1*10^{-2}$ с

для морского района А3 и А4 $t_{\text{дост}}$ составит $11*10^{-2}$ с

Учитывая частоту передачи динамической информации от 2 с до 3 мин, зададим $T_{\text{доп}}$ как 2 с. Такое время позволит отслеживать перемещение и все маневры объекта, что не повлияет на скорость принятия решения судоводителем и в то же время не перегрузит эфир излишними передачами при медленном перемещении корабля.

Определим ограничения по времени обработки информации на АРМ, для каждого морского района:

для морского района А1 $t_{\text{обр}}$ составит $194,97*10^{-2}$ с

для морского района А2 $t_{\text{обр}}$ составит $194,90*10^{-2}$ с

для морского района А3 и А4 $t_{\text{обр}}$ составит $189*10^{-2}$ с

3.3 Определение полного списочного состава

Определение полного списочного состава с техническими характеристиками аппаратуры передачи данных для формирования элементов интегрированной ГИС поддержки принятия решения управления кораблем, таких как: канал связи и состав элементов АРМ (процессор, жесткий диск, оперативная память) удовлетворяющего требованию по времени, с учетом приведенные во второй главе математических моделей.

Из директивных требований к обработке пространственных данных определяется полный списочный состав с техническими характеристиками аппаратуры передачи данных, канала связи и состава элементов АРМ.

В случае если в списочном составе имеется по одному наименованию элементов, то этот список является единственным сценарием для комплексования аппаратуры, удовлетворяющим условию.

В обратном случае, проводится оптимизация структуры по стоимости и эргономике.

Так для океанографического исследовательского судна «Адмирал Владимирский» учитывая все вышесказанное, полный списочный состав будет следующий:

Таблица 3.1. Список процессоров и оперативной памяти

Название	Наименование	Частота, ГГц	Средняя стоимость, руб	Тип ОЗУ	Частота	Производитель	Объем, ГБ
A1	Intel Xeon E5-2670 V3	2,3	125790	DDR 3	2133	Kingston HyperX	8
A1	Intel Xeon E5-2670 V3	2,3	125790	DDR 3	2133	Kingston HyperX Savage	4

A1	Intel Xeon E5-2670 V3	2,3	125790	DDR 4	2133	Samsung	16
A1	Intel Xeon E5-2670 V3	2,3	125790	DDR 4	2133	Corsair	8
A1	Intel Xeon E5-2670 V3	2,3	125790	DDR 4	2133	Patriot Signature	4
A1	Intel Xeon E5-2660 V4	2	114090	DDR 4	2400	Patriot Signature	16
A1	Intel Xeon E5-2660 V4	2	114090	DDR 4	2400	Corsair	32
A1	Intel Xeon E5-2660 V4	2	114090	DDR 3	2400	Kingston HyperX	8
A1	Intel Xeon E5-2660 V4	2	114090	DDR 3	2400	Kingston HyperX	4
A1	Intel Xeon E5-2660 V4	2	114090	DDR 4	2400	Patriot Signature	4
A1	Intel Xeon E5-2660 V4	2	114090	DDR 4	2400	Corsair	8
A1	Intel Xeon E5-2650 V4	2,2	93260	DDR 4	2400	Patriot Signature	16
A1	Intel Xeon E5-2650 V4	2,2	93260	DDR 4	2400	Corsair	32
A1	Intel Xeon E5-2650 V4	2,2	93260	DDR 3	2400	Kingston HyperX	8
A1	Intel Xeon E5-2650 V4	2,2	93260	DDR 3	2400	Kingston HyperX	4
A1	Intel Xeon E5-2650 V4	2,2	93260	DDR 4	2400	Patriot Signature	4
A1	Intel Xeon	2,2	93260	DDR	2400	Corsair	8

	E5-2650 V4			4			
A1	Intel Xeon E5-2630 V4	2,2	54090	DDR 3	2133	Kingston HyperX	8
A1	Intel Xeon E5-2630 V4	2,2	54090	DDR 3	2133	Kingston HyperX Savage	4
A1	Intel Xeon E5-2630 V4	2,2	54090	DDR 4	2133	Samsung	16
A1	Intel Xeon E5-2630 V4	2,2	54090	DDR 4	2133	Corsair	8
A1	Intel Xeon E5-2630 V4	2,2	54090	DDR 4	2133	Patriot Signature	4
A1	Intel Xeon E5-2650 V3	2,3	90120	DDR 3	2133	Kingston HyperX	8
A1	Intel Xeon E5-2650 V3	2,3	90120	DDR 3	2133	Kingston HyperX Savage	4
A1	Intel Xeon E5-2650 V3	2,3	90120	DDR 4	2133	Samsung	16
A1	Intel Xeon E5-2650 V3	2,3	90120	DDR 4	2133	Corsair	8
A1	Intel Xeon E5-2650 V3	2,3	90120	DDR 4	2133	Patriot Signature	4
A1	Intel Xeon E5-2620 V4	2,1	31386	DDR 3	2133	Kingston HyperX	8
A1	Intel Xeon E5-2620 V4	2,1	31386	DDR 3	2133	Kingston HyperX Savage	4
A1	Intel Xeon	2,1	31386	DDR	2133	Samsung	16

	E5-2620 V4			4			
A1	Intel Xeon E5-2620 V4	2,1	31386	DDR 4	2133	Corsair	8
A1	Intel Xeon E5-2620 V4	2,1	31386	DDR 4	2133	Patriot Signature	4

Таблица 3.2 Список внешних запоминающих устройств

Код_зоны	Название	Наименование	Объем, Тб	Средняя цена
1	A1	Seagate Constellation ES.3 [ST2000NM0033]	2	8799
1	A1	Seagate Desktop SSHD [ST4000DX001]	4	11999
1	A1	WD Caviar Red [WD50EFRX]	5	17499
1	A1	WD Caviar Red [WD60EFRX]	6	19299
1	A1	Seagate Enterprise NAS [ST6000VN0001]	6	21999
1	A1	SSD Samsung 850 Pro [MZ-7KE1T0BW]	1,24	34499

Таблица 3.3 Список компонентов локально вычислительной сети

Зо- ны.Название	ШИ- НА(сеть).Название	Сред- няя цена	Наименова- ние	Количе- ство пор- тов	Сред- няя цена, руб
A1	RG-213	82	Cisco SRW208G	8	17563

A1	RG-213	82	Cisco Catalyst 2960-24TT	24	41990
A1	RG-213	82	Cisco SLM2008T	8	7860
A1	RG-6U+CU	13	Cisco SRW208G	8	17563
A1	RG-6U+CU	13	Cisco Catalyst 2960-24TT	24	41990

Поскольку в списочном составе имеется не по одному наименования, то проведем оптимизацию структуры по стоимости и эргономике. Тогда интегрированная геоинформационная система поддержки принятия решения управления кораблем будет состоять из следующих элементов:

- УКВ радиостанция
- Автоматическая идентификационная система (АИС)
- Приемник ЦИВ
- Приемник НАВТЕКС
- АРБ «Коспас-Сарсат»
- 3 АРМ:
- Процессор Intel Xeon E5-2620 V4, 2,1ГГц
- Оперативная память Patriot Signature, 4 ГБ
- Два внешних запоминающих устройства Seagate Constellation ES.3,
2Тб
- Маршрутизатор Cisco SLM2008T
- Шина RG-6U+CU

При невыполнении директивных требований к передаче данных и обработке информации применяется характеристическое преобразование Лапласа-

Стилтьеса, для определения «узкого» место маршрута от источника i к приемнику j . Это «узкое место» является причиной увеличения времени передачи данных.

3.4 Шаги методики проектирования структурно-функциональной модели ГИС корабля

Подробное описание необходимых этапов методики проектирования структурно-функциональной модели ГИС корабля приведена в разд. 3.1.1-3.1.3. Последовательность выполнения этапов и их взаимосвязь может быть представлено в виде следующих шагов:

1. Анализ исходных данных
 - 1.1. Определение минимального количества АРМ исходя из назначения и категории корабля
 - 1.2. Определение минимального состава радиооборудования в зависимости от морского района плавания
 - 1.3. Определение количества узлов сети в зависимости от морского района плавания
2. Расчет временных характеристик
 - 2.1. Определение временных характеристик передачи данных до приемника в зависимости от морского района плавания
 - 2.2. Определение временных характеристик установления соединения, учитывая модель установления соединения
 - 2.3. Определение временных характеристик обработки информации для каждого морского района плавания.
 - 2.4. Определение времени доставки данных до лица принимающего решение при управлении кораблем исходя из пп 1.4, 1.5, 1.6
 - 2.5. Проверка условия ограничения на время доставки, рекомендуемые стандартами распространения пространственных данных
3. Определение полного списочного состава

3.1. Определение модели и количества коммутаторов как компонента локально вычислительной сети, исходя из директивных характеристик установления соединения, рассчитанных в пп 2.2.

3.2. Определение вида коаксиального кабеля, исходя из директивных характеристик установления соединения, рассчитанных в пп 2.2

3.3. Определение списка моделей процессоров, исходя из директивных характеристик обработки информации, рассчитанных в пп 2.3

3.4. Определение списка моделей ОЗУ, поддерживаемые процессорами

3.5. Определение списка моделей жесткого диска, , исходя из директивных характеристик обработки информации, рассчитанных в пп 2.3

3.6. Оптимизация списка оборудования обработки информации на АРМ по стоимостным характеристикам

4. При невыполнении условия ограничения на время доставки, рекомендуемые стандартами распространения пространственных данных

4.1. Определение узкого места маршрута, исходя из модели узкое место

4.2. Рекомендации замены узкого места на другой узел с лучшими характеристиками

Выводы по третьей главе

Предложена вероятностная методика проектирования структурно-функциональной модели ГИС корабля, корабля, которая может быть инструментом, позволяющим решать задачу формирования облика модулей ГИС поддержки принятия решения при управлении кораблем. В методике учтены стандарты и требования в судовождении, а так же математические и концептуальная модели из первой и второй главе.

Предложенная методика отличается комбинированным применением этапов автоматической генерации вариантов архитектуры ГИС и экспертных оценок по выбору программно-аппаратных средств ее построения.

В методике описаны этапы ее формирования и пошаговый алгоритм действий для формирования структурно-функциональной модели ГИС корабля

4 Экспертная система принятия решения по выбору структуры ГИС корабля

Экспертная система (ЭС) – это программный инструментарий, который оперирует со знаниями в определенной предметной области с целью выработки рекомендаций (сценариев) или решения проблем. Экспертные системы решают задачи так, как их решил бы специалист или эксперт в данной проблемной области. ЭС, как и эксперт-человек, в процессе своей работы оперирует со знаниями. Экспертные системы выдают советы, проводят анализ, выполняют классификацию, дают консультации и ставят диагноз.

Знания о предметной области определённым образом формализованы и представлены в памяти ЭВМ в виде базы знаний. База знаний может изменяться и дополняться в процессе развития экспертной системы.

Основное отличие компьютерных программ от экспертной системы заключается в том, что программа использует процедурный анализ, а экспертные системы решают задачи с использованием дедуктивных рассуждений. Поэтому ЭС создаются для решения задач в узких проблемных областях.

Достоинства экспертных систем:

– Экспертная система может накапливать знания, таким образом появляется опыт.

– ЭС может сохранять знания длительное время,

– ЭС может обновлять знания.

Эти достоинства позволяют экспертной системе долго находиться в актуальном состоянии и быть независимой от квалифицированных специалистов.

Экспертная система соответствует технологии инженерии знаний, если решаемая с ее помощью задача имеет следующие характеристики:

1) Задача должна решаться посредством манипуляции символами (т.е. с помощью символических рассуждений).

2) Задача должна иметь не алгоритмическую, а эвристическую природу, т.е. с помощью эвристик. Задачи, которые могут быть гарантировано, решены с помощью набора формальных процедур, не подходят для применения ЭС.

3) Задача должна быть достаточно сложной, чтобы ЭС могла ее решить (решение эксперта занимает часы, а не недели).

4) Задача должна быть достаточно узкой и практически значимой для решения ее по технологии инженерии знаний.

По степени сложности решаемых задач экспертные системы можно классифицировать следующим образом (рисунок 4.1):[35]

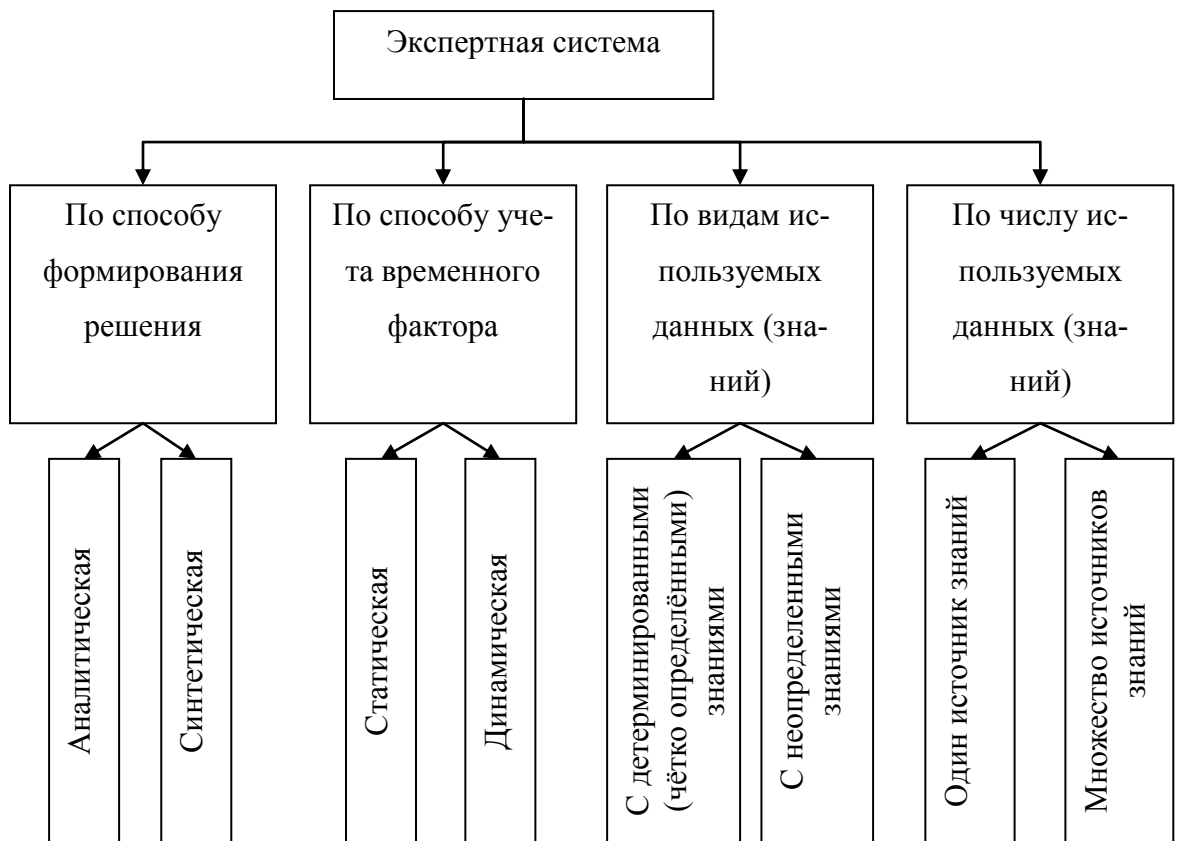


Рисунок 4.1 – Классификация ЭС

Аналитические системы предполагают выбор решений из множества известных альтернатив (определение характеристик объектов), а синтетические системы – генерацию неизвестных решений (формирование объектов).

Статические системы решают задачи при неизменяемых в процессе решения данных и знаниях, динамические системы допускают такие изменения.

Статические системы осуществляют монотонное непрерываемое решение задачи от ввода исходных данных до конечного результата, динамические системы предусматривают возможность пересмотра в процессе решения полученных ранее результатов и данных.

Под неопределенностью знаний (данных) понимается их неполнота (отсутствие), недостоверность (неточность измерения), двусмысленность (многозначность понятий), нечеткость (качественная оценка вместо количественной).

Источники знаний могут быть альтернативными (множество миров) или дополняющими друг друга (кооперирующими).[35,38]

Наиболее известная ЭС, которая определяет местоположение и типы судов в тихом океане по данным акустических систем слежения – интерпретирующая система HASP/SIAP.

В соответствии с перечисленными признаками классификации можно выделить следующие четыре основные класса экспертных систем:

1. Классифицирующая, характеризующаяся решением задачи распознавания ситуаций. В такой системе нахождение одного наилучшего решения из альтернативных, удовлетворяющего поставленной цели и ограничениям, определяется по набору заданных признаков (факторов).

2. Доопределяющая система, в которой решаются более сложный класс аналитических задач на основе неопределенности исходных данных и применяемых знаний.

3. Трансформирующая ЭС, предполагающая повторяющиеся преобразования знаний в процессе решения задач. В этих системах на основании исходных данных осуществляется генерация гипотез, а затем осуществляется проверка сформулированных гипотез на подтверждение поступивших фактов.

4. Многоагентные являются динамическими системами, в которых БЗ сформулирована из нескольких разнородных источников знаний, которые обмениваются между собой результатами на динамической основе.

Разработка ЭС имеет существенные отличия от разработки обычного программного продукта. В ходе работ по созданию ЭС сложилась определен-

ная технология их разработки, включающая шесть следующих этапов, показанных на рисунке 4.2.[38]



Рисунок 4.2 – Стадии проектирования ЭС

На этапе **идентификации** определяется, какие задачи будет решать ЭС в будущем. Формируются требования к ЭС. Определяются участники процесса проектирования ЭС и их роли. Результатом этого шага является ответ на вопрос, что необходимо делать и какие результаты надо получить.

В процессе идентификации задачи инженер по знаниям и эксперт работают в тесном контакте. Сначала эксперт неформально делает описание задачи. Затем инженер по знаниям уточняет термины и ключевые понятия. После этого эксперт корректирует описание задачи и объясняет, как решать ее и какие рассуждения лежат в основе того или иного решения. После нескольких уточняющих циклов эксперт и инженер по знаниям получают окончательное неформальное описание задачи. [71]

Ресурсами проектирования ЭС являются источники знаний, время разработки, вычислительные средства и объем финансирования. Для эксперта источниками знаний служат его предшествующий опыт по решению задачи, книги, известные примеры решения задач.

Для инженера по знаниям источником знаний является опыт решения аналогичных задач, методы представления знаний и манипулирования ими, программные инструментальные средства

На этапе **концептуализации** проводится содержательный анализ проблемной области, выявляются понятия и их взаимосвязи, определяются методы решения задач. Этот этап заканчивается созданием модели предметной области (ПО), включающей основные концепты и отношения. На этапе концептуализации определяются следующие особенности задачи: типы данных; исходные и выводимые данные, подзадачи общей задачи; используемые стратегии и гипотезы; виды взаимосвязей между объектами ПО, типы используемых отношений (иерархия, причина - следствие, часть - целое и т.п.); процессы, используемые в ходе решения; состав знаний, используемых для решения задачи.[55,56]

На этапе **формализации** все ключевые понятия и отношения выражаются на некотором формальном языке. Другими словами, на данном этапе определяются состав средств и способы представления фактов и правил. Результатом этапа формализации является описание того, как задача может быть представлена в выбранной модели представления знаний и определение способов манипулирования этими знаниями и интерпретации знаний[56,63]

Цель этапа **выполнения** - создание одного или нескольких прототипов ЭС, решающих требуемые задачи. Затем на данном этапе по результатам тестирования и опытной эксплуатации создается конечный продукт, пригодный для промышленного использования. Разработка прототипа состоит в программировании его компонентов или выборе их из известных инструментальных средств и наполнении базы знаний[63]

На этапе **тестирования** инженер по знаниям подбирает примеры, обеспечивающие проверку всех возможностей разработанной экспертной системы.[]

На этапе **опытной эксплуатации** проверяется пригодность ЭС для пользователя. Пригодность означает удобство работы с ЭС и полезность ЭС.

4.1 Общая характеристика экспертной системы

Экспертная система принятия решения по выбору структуры ГИС корабля послужит программной реализацией инженерной методики проектирования ГИС корабля. Разработка и внедрение методики представляются достаточно актуальными, так как это позволит значительно сократить количество времени, затрачиваемого на определение аппаратного облика ГИС корабля, и повысит эффективность работы ГИС.

Экспертная система, разрабатываемая в диссертационной работе, предназначена для формирования облика аппаратного слоя ГИС. Такое решение позволяет реализовать не только простые средства общения между системой и специалистами, но и довести хранимые в системе знания до специалиста вместе с необходимыми пояснениями и разъяснениями.

Принцип работы данной экспертной системы позволяет описать каждый сценарий в виде набора вопросов, ответы на которые нужно получить от пользователя, чтобы определить, какой именно вариант комплексирования элементов ЛВС и навигационного оборудования удовлетворяет заданному требованию. В результате система формирует оптимальный списочный состав оборудования, учитывающий нормативные документы и директивные характеристики.

Типичная ЭС состоит из следующих основных компонентов

- решателя (интерпретатора);
- рабочей памяти (РП), называемой также базой данных (БД);
- базы знаний (БЗ);
- компонентов приобретения знаний;
- объяснительного компонента;
- диалогового компонента.[35]

Теперь, когда мы знаем, из каких компонент в общем состоит экспертная система, построим диаграмму, отражающую структуру такой экспертной системы (Рисунок 4.3).

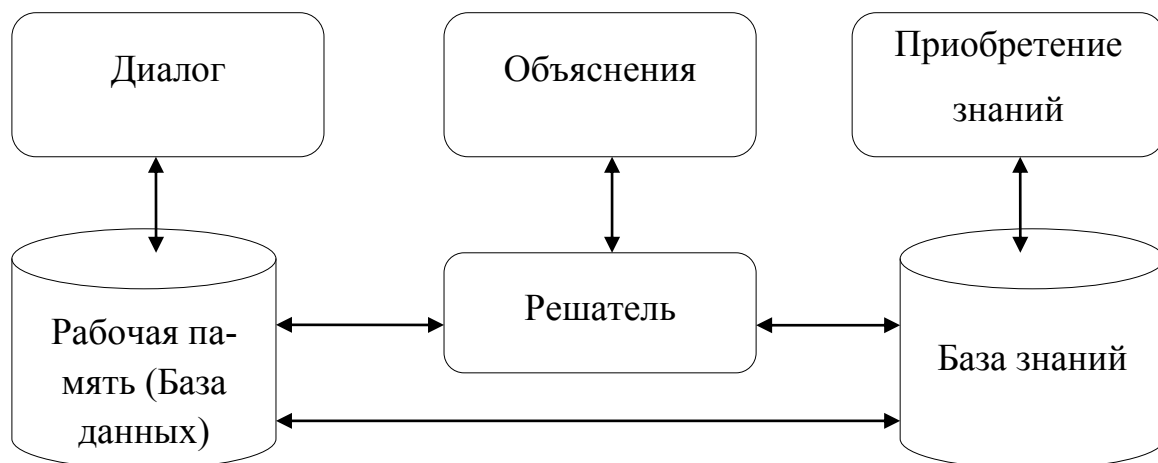


Рисунок 4.3 – Структура экспертной системы

База данных (рабочая память) предназначена для хранения исходных и промежуточных данных задачи, которая решается в текущий момент времени.

База знаний в ЭС предназначена для хранения долгосрочных данных, описывающих свойства проблемной области и правил, описывающих преобразования данных.

Решатель, использует исходные данные из рабочей памяти и знания из БЗ и формирует правила. Затем исходные данные подставляются в эти правила и достигается цель задачи.[35]

Компонент приобретения знаний необходим для автоматизации наполнения экспертной системы знаниями от эксперта.

Объяснительный компонент объясняет, как система получила решение задачи и какие знания она при этом использовала. Наличие этого компонента повышает доверие пользователя к полученному результату.

Диалоговый компонент реализует дружественное общение экспертной системы с пользователем как в ходе решения задач, так и в процессе приобретения знаний и объяснения результатов работы.

Поскольку для решения данной ЭС предполагается использование сценарного подхода, рассмотрим алгоритмы поиска решений, которые применяются в сценарном моделировании.

Прежде всего, необходимо выбрать представление, в котором реализуются процедуры поиска решений и организации вычислительного процесса. В качестве такого представления целесообразно выбрать представление в пространстве состояний. Тогда задача поиска решений формально описывается следующим образом: $T = \langle S, S_0, S_k, F \rangle$, где S_0 – начальное состояние, S_k – конечное состояние или состояния, S – множество промежуточных состояний, $F = \{F^{\eta}_i\}$ – множество операторов, которые переводят процесс поиска из одного состояния в другое. Каждому математическому отношению F_i поставим в соответствие список (кортеж) входящих в него параметров. Таким образом, рассматриваемые алгоритмы предусматривают работу со списочными структурами данных.

При поиске решений в сценарном моделировании в качестве множества операторов выступают разрешения математических отношений F^{η}_i , реализуемые в виде отдельных программных модулей, совокупность которых для данной проблемной области составляет локальную (быть может, одну из многих) базу процедур. Здесь верхний индекс η указывает на параметр, который в данном разрешении выступает как функция, а нижний индекс i – на номер соответствующего математического отношения в совокупности математических отношений. Задание исходных данных определяет начальное состояние S_0 , а искомое решение – конечное (целевое) состояние. Выбор на каждом очередном шаге некоторого конкретного оператора осуществляется в соответствии с некоторыми правилами, которые для данной проблемной области составляют локальную базу правил.[35]

Алгоритм 1. Стратегия обратной волны.

В алгоритме обратной волны реализуется поиск решения задачи с целевого состояния, т.е. от искомого параметра. Суть алгоритма состоит в следующем. В соответствии с алгоритмом поиска решений образуются следую-

щие списки: S_1 – список параметров, которые должны быть рассчитаны; S_2 – список параметров, для которых выбраны разрешения для расчета; дополнительно образуем еще два списка: S_3 – список разрешений, включаемых в план решения задачи; S_4 – список оценок сложности реализации разрешения, выбранного в план решения задачи. Данные оценки позволяют при наличии нескольких планов выбрать наилучший, т. е. реализовать классическую постановку задачи принятия решений.[35]

Шаг 1. Поместить искомый параметр в список S_2 и найти разрешение, в дальнейшем будем называть оператор F^n_i , с помощью которого может быть рассчитан искомый параметр. Если таких разрешений несколько, выбрать разрешение, содержащее наименьшее число неизвестных аргументов. Если все аргументы данной оператора известны, то план решения найден, можно переходить к собственно организации вычислений, иначе - записать неизвестные аргументы в список S_1 и перейти к шагу 2.

Шаг 2. Упорядочить список S_1 в соответствии со следующим правилом: самым первым в списке поместить параметр, для которого разрешение имеет наибольшее число неизвестных аргументов, а на последнее место – параметр, для которого все аргументы известны (если такое разрешение имеет место). Перейти к шагу 3.

Шаг 3. Выбрать очередной неизвестный параметр из списка S_1 и поместить его в список S_2 . Если это был последний в списке параметр и список S_1 стал пустым, то поиск данного плана решения закончен, можно переходить либо к поиску следующего возможного плана, либо к организации вычислений в соответствии с данным планом. Иначе – поместить неизвестные аргументы данного разрешения в список S_1 , и перейти к шагу 2.

Если набор математических отношений является неполным, возможно возникновение ситуаций, когда в рамках данного набора математических отношений и принятых правил выбора не удастся найти маршрут для вычисления требуемого параметра. Таким образом, возникает тупиковая ситуация.

Тупиковая ситуация – это появление стыков и вилок. Стык – когда оператор F_i^n не имеет выхода, а вилка – когда оператор F_i^n не имеет на входе данных. Еще одна ситуация может встретиться, когда в процессе формирования рабочей программы (организации вычислительного процесса) образуются циклы, т.е. для расчета некоторому параметра требуется знать значение этого же параметра.

Алгоритм 2. Стратегия прямой волны.

В алгоритме прямой волны планирование идет от исходных данных к целевому параметру. Суть алгоритма состоит в следующем.

Обозначим: $N_{\text{чп}}$ – число параметров в математическом отношении; D – число заданных исходных данных или значений параметров; $N_{\text{чнп}}$ – число параметров, оставшихся неизвестными; $N_{\text{чип}}$ – число параметров, найденных или известных на данном шаге применением соответствующего оператора F .

Шаг 1. Пометить параметры, значения которых известны, и для каждого F_i (списка параметров) рассчитать значение $N_{\text{чнп}} = N_{\text{чп}} - D_i^j$, где j – номер шага алгоритма и $D_i^{j=0} \leq D$

Шаг 2. Пометить F_i^n , $i \in (1, M)$, где M есть полное множество математических отношений, описывающих проблемную область решаемых задач, для которых $N = 1$; положить в каждом из помеченных через F_i списков параметров неизвестный параметр η как функцию и вычислить его значение во всех F_i

В результате выполнения данного шага множество известных параметров расширится от D до $D_j > D$.

Шаг 3. Перейти к шагу 1.

Поиск решения можно представить графом состояний следующего вида:

$$S_0: \langle a^*, b^*, c^* | d \rangle$$

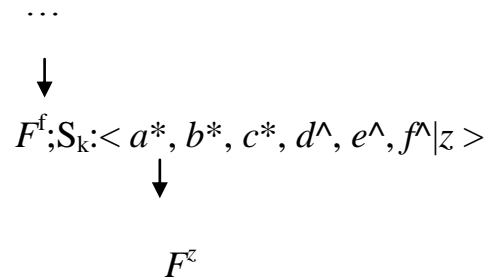


$$F^d; S_1: \langle a^*, b^*, c^*, d^\wedge | e \rangle$$



$$F^e; S_2: \langle a^*, b^*, c^*, d^\wedge, e^\wedge | f \rangle$$





В данном алгоритме за один шаг находятся все разрешения, с помощью которых могут быть рассчитаны какие-либо параметры. Далее пересчитываются значения N_i , и действие повторяется. И так до момента, пока на некотором шаге в число рассчитываемых не попадет целевой параметр F^z .

В общем случае при реализации данного алгоритма могут быть лишние вычисления, а именно вычисления параметров, которые не являются необходимыми при расчете целевого параметра. Тогда, сформулировав некоторые дополнительные правила, можно уменьшить сложность вычислительного алгоритма. Ситуация несколько усложняется в случае задания на расчет не одного, а нескольких целевых параметров.

Как нетрудно заметить, возможна модификация данного алгоритма, когда на каждом шаге определяется первый оператор F_i для которого $N_i^j = 1$, и рассчитывается соответствующий параметр. После этого поиск повторяется.

Сравнение двух алгоритмов стратегий принятия решений в сценарном моделировании показывает, что алгоритм «прямой волны» является мобильным, т.к. за один проход алгоритма могут быть найдены все возможные решения, которые имеются в данной предметной области. При реализации алгоритма «обратной волны» будет найдено одно оптимальное решение и при необходимости определения другого параметра алгоритм должен быть реализован заново.

4.2 Реализация экспертной системы

Экспертная система реализована с использованием среды разработки Delphi и внешней базой данных, являющейся рабочей памятью системы. Она

решает задачу выбора оборудования для формирования структурно-функциональной модели ГИС поддержки принятия решения при управления кораблем.

Работа поиска решения основана на алгоритме прямой волны, планирование в которой идет от исходных данных к целевому параметру.

В процессе диалога запрашиваются исходные данные, и имеется визуализацию решаемой проблемы, в виде карты, отображающей морской район плавания, выбранный пользователем.

Рабочая память (база данных) имеет 9 таблиц, которые при необходимости могут дополняться и редактироваться. Таблицы хранят в себе данные о необходимом навигационном оборудовании наличие, которого определено в требованиях ГМССБ, компонентами для формирования сети и наиболее важных компонентах АРМ. Таблицы со связями представлена на рисунке 4.4

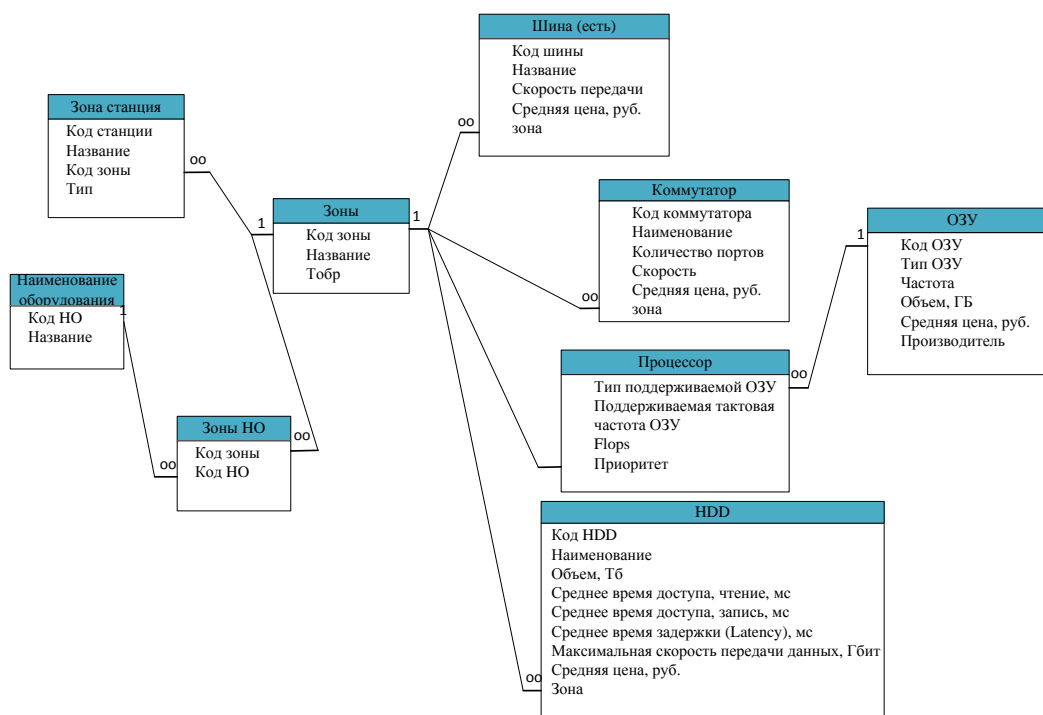


Рисунок 4.4 – Структура рабочей памяти ЭС

База знаний определяет правила работы экспертной системы и связана с рабочей памятью. Она определяет требований к архитектуре ГИС поддержки принятия решения при управлении корабле, имея логическую структуру

Определение количества АРМ

Если <категория корабля>=транспортное то <количество АРМ>=3

Если <категория корабля>=промысловое то <количество АРМ>=3

Если <категория корабля>=НИС то <количество АРМ>=4

Определение количества узлов и времени доставки данных до приемника:

Если <морской район>=A1 то <количество узлов без АРМ>=5, <время доставки>= $32,1 * 10^{-5}$

Если <морской район>=A2 то <количество узлов без АРМ>=9, <время доставки>= $106 * 10^{-5}$

Если <морской район>=A1 то <количество узлов без АРМ>=11, <время доставки>= $0,65 * 10^{-1}$

Если <морской район>=A1 то <количество узлов без АРМ>=13, <время доставки>= $0,65 * 10^{-1}$

Определение сценариев:

Если <время доставки+время установления соединения+время обработки> < <допустимого времени> то < вывести полный списочный состав>

Если <в списочном составе больше одного наименования в категориях> то <оптимизировать по цене>

Если <время доставки+время установления соединения+время обработки> > <допустимого времени> то < <найти узкое место и провести оптимизацию ГА>

Работу базы знаний можно представить в виде дерева решений, представленного на рисунке 4.5

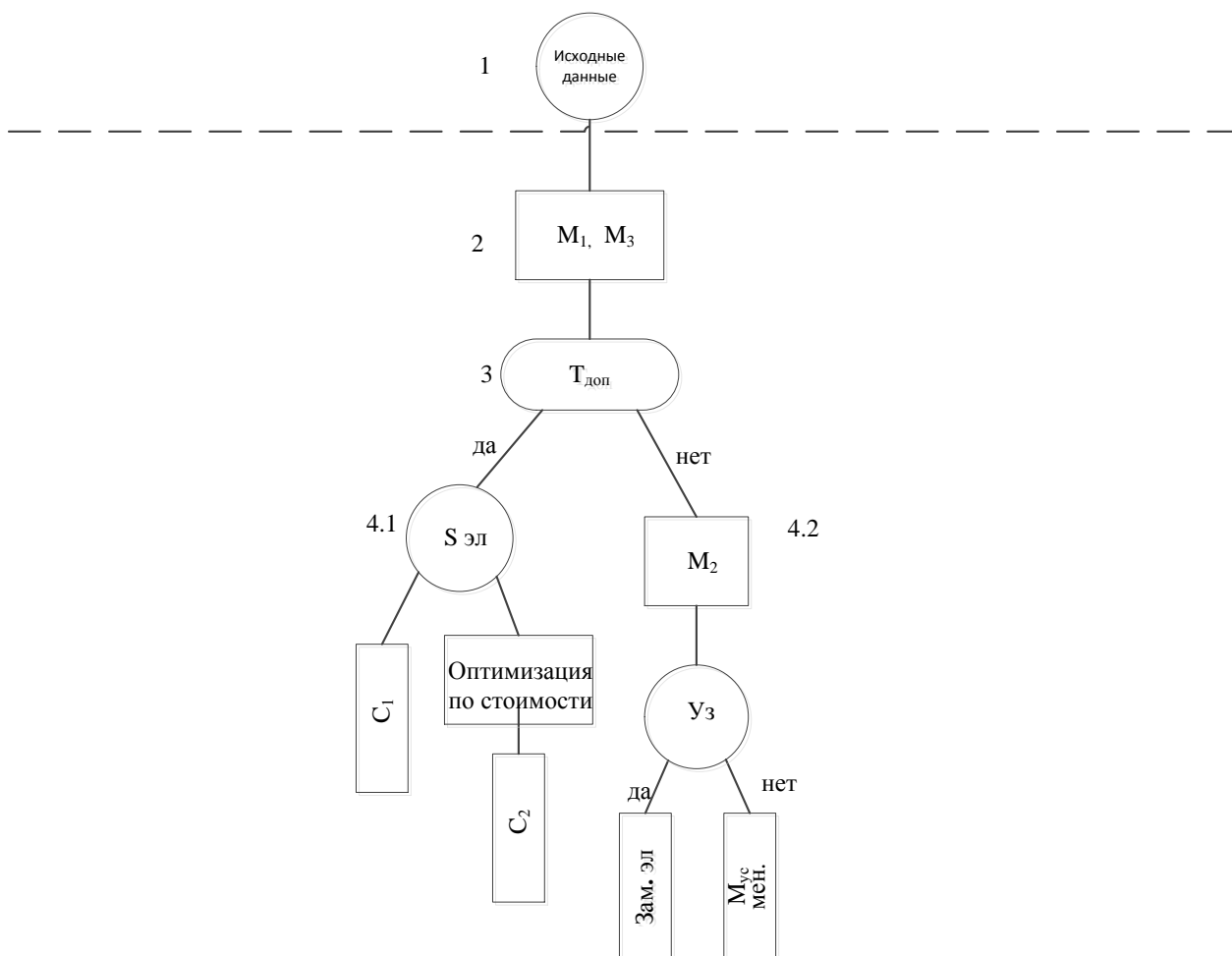


Рисунок 4.5 – Дерево решений ЭС

Решатель, исходя из логики базы знаний, принимает решение о применении моделей описанных во второй главе. После определения типа корабля и морского района плавания определяется количество узлов ГИС. Рассчитывается время доставки данных до системы, применяя модели M_1 , M_2 , M_3 и определяется списочный состав элементов ГИС удовлетворяющих директивному требованию. Если в списочном составе имеется несколько наименований по отдельным категориям, то проводится оптимизация.

В случае, когда при решении не выполняются директивные требования, то производится поиск узкого места используя модель M_2 . Применяется генетический алгоритм для поиска решения задачи.

Схематически работу решателя можно отобразить в виде блок-схемы представленной на рисунке 4.6

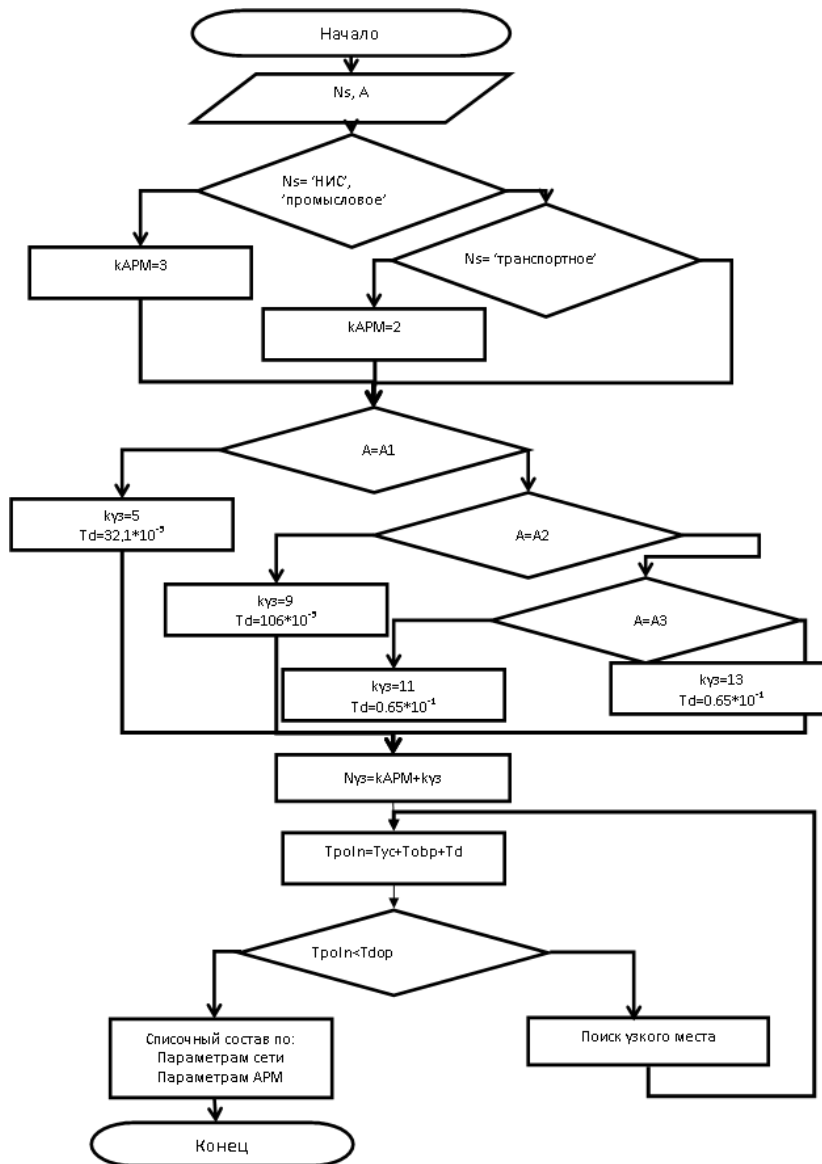


Рисунок 4.6 – Блок-схема работы решателя

Диалог работы с пользователем в виде графического интерфейса. Для начала работы необходимо выбрать категорию транспорта и морской район (Рисунок 4.7)



Рисунок 4.7 – Выбор исходных параметров.

При выборе одного из морских районов плавания на карте выделяется зона, которая относится к данному району. На рисунках 4.8 и 4.9 продемонстрирован выбор морских районов А1 и А2.

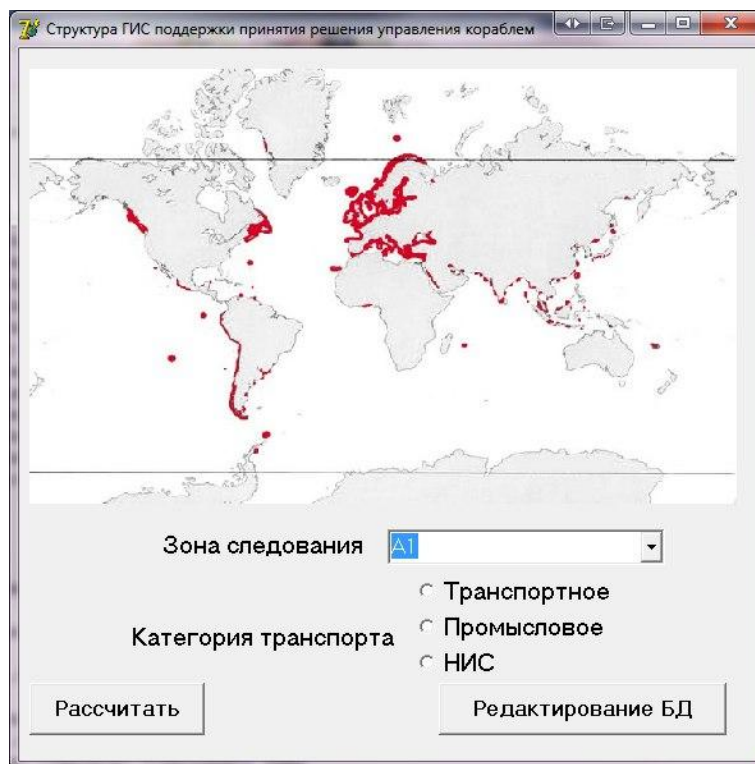


Рисунок 4.8 – Выбор морского района А1

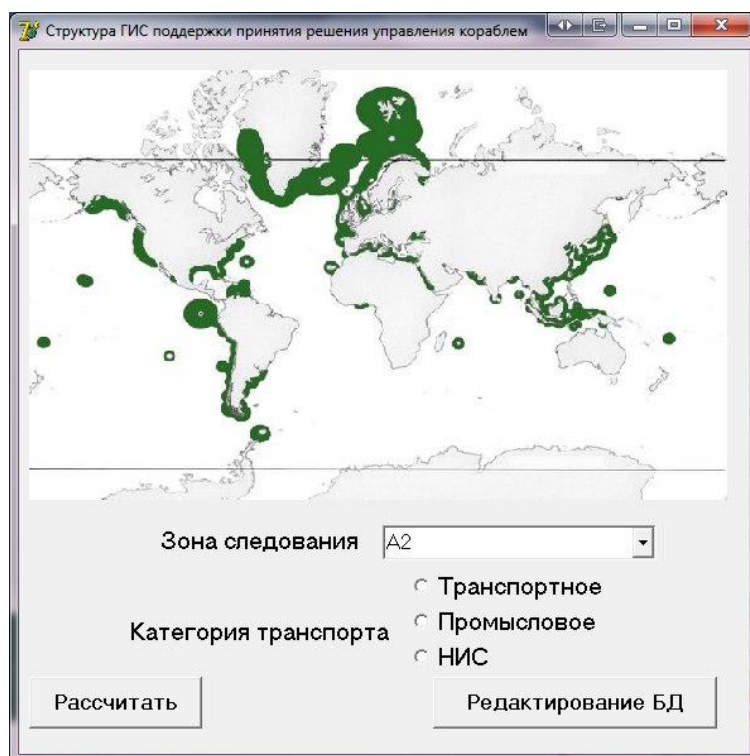


Рисунок 4.9 – Выбор морского района А2

После чего необходимо нажать кнопку «Рассчитать». Программный инструмент обращается к базе знания для определения правил, описанных выше, и из рабочей памяти выбирает варианты удовлетворяющие условиям,

прописанных в логике базы знаний. Далее проводятся все расчеты определенными решателем и выдаются все возможные варианты удовлетворяющие директивным требованиям (Рисунок 4.10, 4.11)

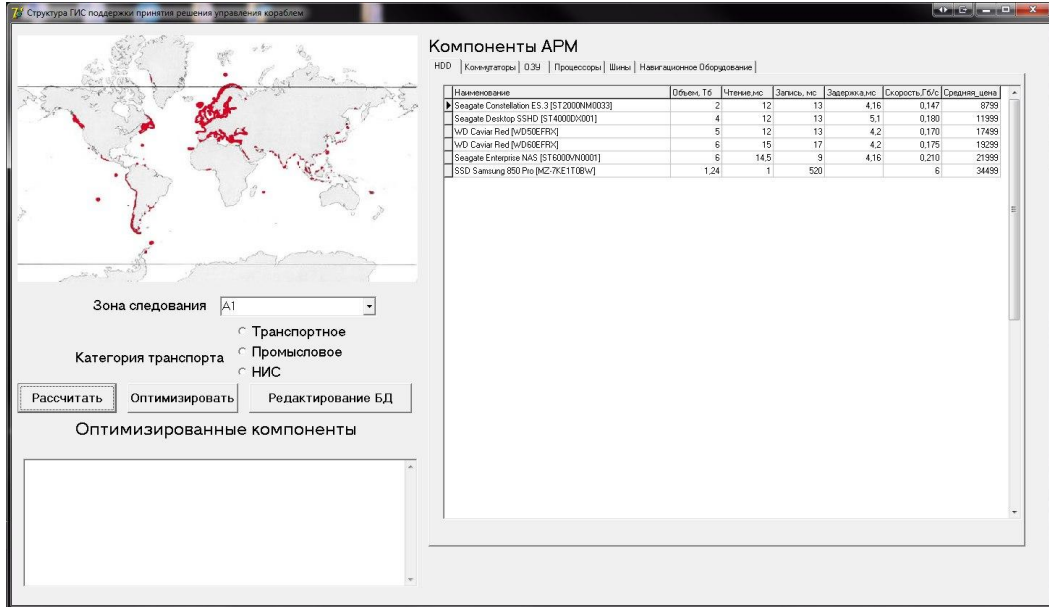


Рисунок 4.10 – Варианты оборудования удовлетворяющие директивным требованиям для морского района А1

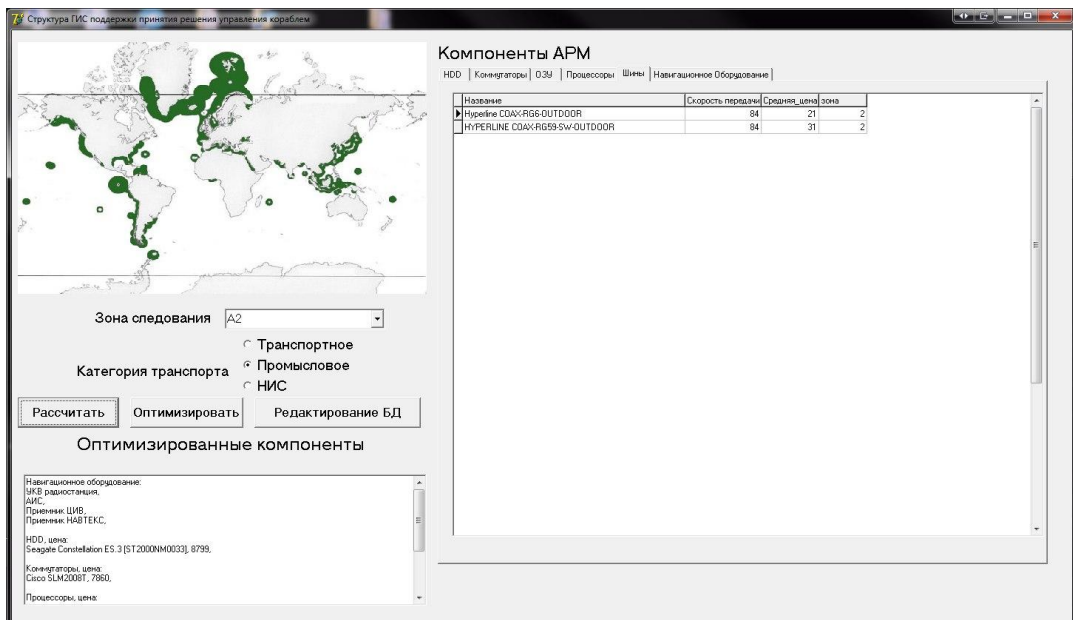


Рисунок 4.11 – Варианты оборудования удовлетворяющие директивным требованиям для морского района А2

Для получения конечного списка оптимизированного по стоимостным и площадным характеристикам необходимо нажать кнопку «Оптимизировать»

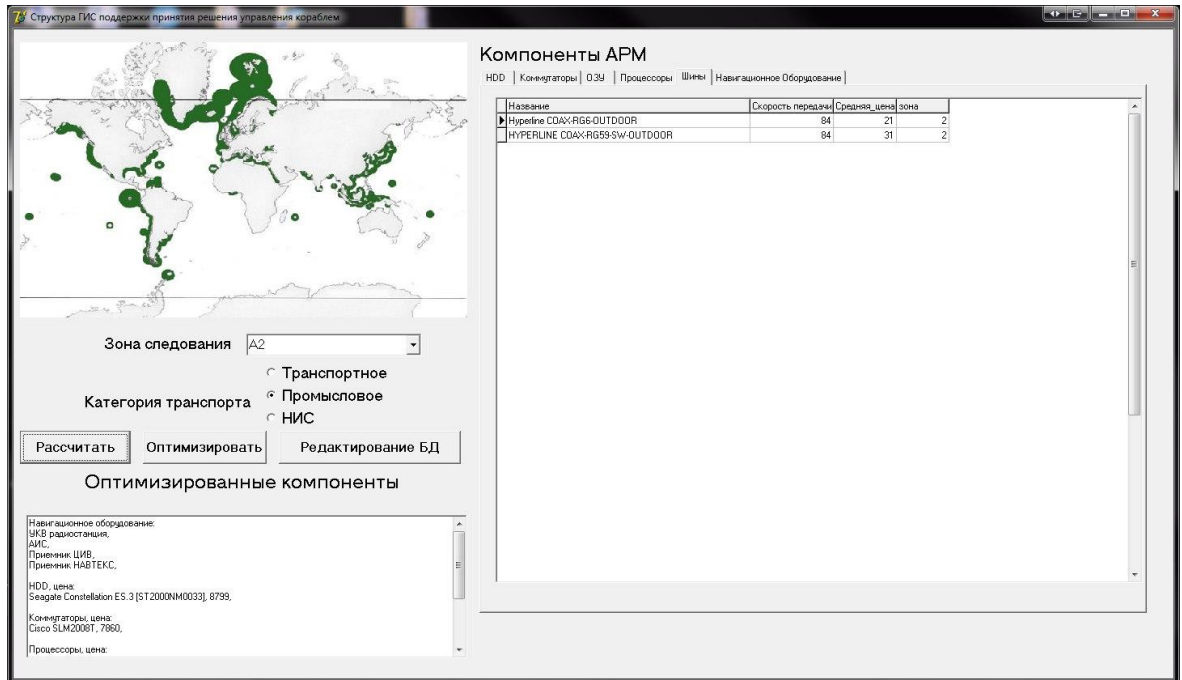


Рисунок 4.12 – Оптимизированный список оборудования удовлетворяющие директивным требованиям для морского района A2

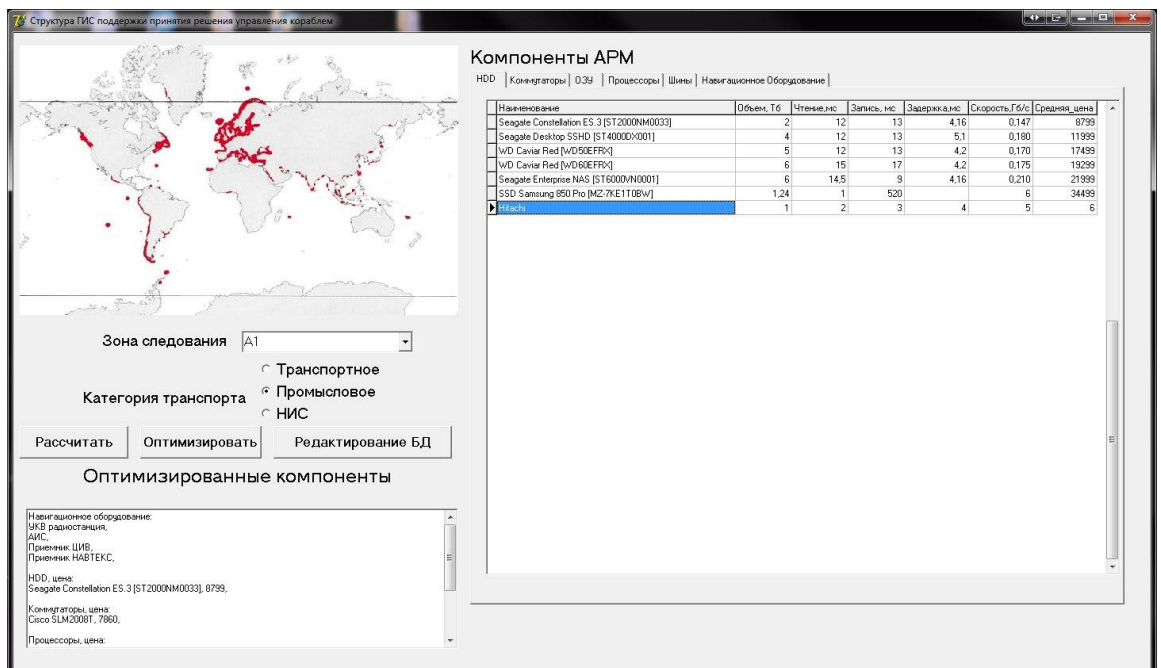


Рисунок 4.13 – Оптимизированный список оборудования удовлетворяющие директивным требованиям для морского района A1

На рисунках 4.12 и 4.13 представлен вариант работы программы после оптимизации для морских районов плавания A1 и A2, категории транспорта «промышленное»

При необходимости эксперт может отредактировать рабочую память, добавив или удалив записи в базе данных. Для этого необходимо нажать кнопку «Редактировать БД» (Рисунок 4.14)

The screenshot shows a software window titled 'Form2' with a menu bar containing 'HDD', 'Компьютеры', 'ОЗУ', 'Процессоры', 'Шины', and 'Навигационное Оборудование'. The main area displays a table of HDD specifications. On the left side, there are input fields for various parameters and buttons for 'Обновить', 'Добавить', 'Удаление', and 'Назад'.

Код_HDD	Наименование	Объем, Тб	чтение, мс	запись, мс	задержки, мс	Скорость, Гбит/с	Средняя цена	зона
5	Seagate Constellation ES.3 [ST1000NM0033]	1	9,5	10,5	4,16	0,175	6799	2
6	Toshiba P300 [HDW120LZSVA]	2	8,5	9,5	4,2	0,190	4750	3
7	Seagate Barracuda [ST2000DM001]	2	8,5	9,5	4,16	0,125	4999	3
8	WD Blue [wD20EZRZ]	2	11	12	5,5	0,147	5299	2
9	Seagate Constellation ES.3 [ST2000NM0033]	2	12	13	4,16	0,147	8799	1
10	WD Caviar Red Pro [wD2001FFSK]	2	6,5	6,5	4,17	0,164	9999	4
11	Toshiba E300 [HDWA130LZSVA]	3	4,2	5,4	5,06	0,196	6099	4
12	WD Caviar Green IntelliPower [wD30EZRK]	3	8,9	10,9	4,2	0,147	6799	3
13	WD Blue [wD30EZRZ]	3	11	12	5,5	0,147	7399	2
14	Seagate Constellation ES.3 [ST3000NM0033]	3	9,5	10,5	4,16	0,165	10999	2
15	WD Caviar Red Pro [wD3001FFSK]	3	8,5	8,5	4,16	0,168	13299	3
16	WD Blue [wD40EZRZ]	4	11	12	5,5	0,150	9699	2
17	Seagate Surveillance [ST4000VX000]	4	8,5	9,5	4,16	0,180	10299	3
18	Seagate Desktop SSHD [ST4000D001]	4	12	13	5,1	0,180	11999	1
19	Seagate Constellation ES.3 [ST4000NM0033]	4	8,5	9,5	4,16	0,170	14299	3
20	WD Caviar Black [wD4001FAEX/wD4003FZEK]	4	8,5	9,5	4,2	0,171	15999	3
21	WD Caviar Red [wD50EFRK]	5	12	13	4,2	0,170	17499	1
22	Seagate Surveillance [ST5000VX0001]	5	8,5	9,5	4,16	0,180	18999	3
23	Surveillance [ST6000VX0001]	6	8,5	9,5	4,16	0,209	17699	3
24	WD Blue [wD60EZRZ]	6	11	12	5,5	0,175	17699	2
25	WD Caviar Red [wD60EFRK]	6	15	17	4,2	0,175	19299	1
26	Seagate Enterprise NAS [ST6000VX0001]	6	14,5	9	4,16	0,210	21999	1
27	Seagate Archive HDD [ST6000AS0002]	8	11	5,5	4,16	0,250	20999	2
28	Hitachi Ultrastar He8 [HUH728080ALE604]	8	8,5	9	4,2	0,205	33499	3
29	Hitachi Ultrastar He10 [HUH721010ALE604]	10	8	8,5	4,16	0,250	39999	3
30	SSD Intel 540s [SSDS5C2Kw010x6K1]	1	1	480	6		22999	3
31	SSD Samsung 850 EVO [MZ75E1T0Bw]	1	1	520	6		25466	4
33	SSD Plextor M6 Pro [PX-1TM6Pro]	1,24	1	490	6		29999	4
34	SSD Samsung 850 Pro [MZ7KE1T0Bw]	1,24	1	520	6		34499	1
35	SSD Crucial M300	2	1	510	6		43999	4
44	WD	12	13	14	151	16	1777	2
133	Hitachi	1	2	3	4	5	6	1

Рисунок 4.14 – Пример редактирования базы данных

Выводы по четвертой главе

Предложена экспертная система реализующая методика проектирования инфраструктуры ГИС корабля, которая может быть инструментом, позволяющим решать задачу по формированию сценариев необходимого оборудования в зависимости от исходных данных

В экспертной системе учтены, стандарты и ГОСТы для определения наличия обязательного оборудования входящих в ГИС, а так же система математических модели по оценке производительности ГИС корабля на всех этапах информационного взаимодействия.

Произведена программная реализация экспертной системы, позволяющая в диалоговом режиме определить исходные данные, для дальнейшего формирования облика ГИС поддержки принятия решения при управлении кораблем

Заключение

В работе были получены следующие основные результаты:

1. Обоснована необходимость интеграции оборудования в единую геоинформационную инфраструктуру для возможности перехода от функционально-специализированных архитектур средств обработки сигналов и данных к распределенной сетевой обработке, обеспечения взаимодействия комплексов, станций, систем и функциональных элементов не на основе парных связей, а на основе базовой информационной транспортной сети корабля и обеспечения возможности подключения к базовой корабельной сети комплексов, станций, систем и функциональных элементов

Автором лично предложена **концептуальная модель ГИС корабля**, представленная на уровне информационных ресурсов, программного обеспечения и технических средств, отражающих функциональное назначение ГИС и ее архитектуру, которая отличается описанием иерархии компонентов, поддерживающих функциональность ГИС и представлением минимально полного набора компонентов ГИС корабля.

2. Разработана **система математических моделей** по оценке производительности ГИС корабля для всех этапов информационного взаимодействия, который отличается сочетанием аналитического и статистического методов моделирования на всех этапах прохождения сигнала. Проведены эксперименты на моделях и получены соответствующие результаты.

3. Разработана **методика проектирования структурно-функциональной модели ГИС**, предназначенная для решения задачи формирования инфраструктуры ГИС поддержки принятия решения при управлении кораблем, которая включает в себя приближение архитектуры ГИС к заданному набору свойств и последовательность действий для приближения архитектуры ГИС. Приближение архитектуры ГИС к заданному набору свойств сводится к выбору из множества параметров, которые задаются как технические требования на составные элементы ГИС, наилучших. Автором подробно описаны необходимые этапы методики проектирования структурно-

функциональной модели ГИС корабля и был предложен пошаговый алгоритм проектирования структурно-функциональной модели ГИС корабля, отличающийся комбинированным применением этапов автоматической генерации вариантов архитектуры ГИС.

4. Разработана **экспертная система по выбору облика ГИС корабля** в рамках верификации методики проектирования структурно-функциональной модели геоинформационной системы корабля, которая позволяет в диалоговом режиме формировать списочный состав модулей необходимых для проектирования, а также вносить изменения в рабочую память. Автором лично было реализовано взаимодействие решателя с рабочей памятью, на основе логики прописанной в базе знаний и применены математические модели по оценке производительности геоинформационной системы корабля на всех этапах информационного взаимодействия

Литература

1. ГОСТ 21063-81 Оборудование навигационное судовое. Термины и определения
2. ГОСТ 25792-85 Приемники морской подвижной службы. Параметры, общие технические требования и методы измерений
3. ГОСТ 26897-86 Радиостанции с однополосной модуляцией морской подвижной службы. Типы, основные параметры, технические требования и методы измерений
4. ГОСТ Р МЭК 60945-2007 Морское навигационное оборудование и средства радиосвязи. Общие требования. Методы испытаний и требуемые результаты испытаний
5. ГОСТ 16504-81 Система государственных испытаний продукции. Испытания и контроль качества продукции. Основные термины и определения
6. ГОСТ 30709-2002 Техническая совместимость. Термины и определения
7. ГОСТ Р ИСО/МЭК 27001-2006 Информационная технология. Методы и средства обеспечения безопасности. Системы менеджмента информационной безопасности. Требования
8. ГОСТ Р 52611-2006 Системы промышленной автоматизации и их интеграция. Средства информационной поддержки жизненного цикла продукции. Безопасность информации. Основные положения и общие требования
9. ГОСТ 19176-85 Системы управления техническими средствами корабля. Термины и определения
10. Алиев Т. И. Проектирование систем с приоритетами // Известия высших учебных заведений. Приборостроение. 2014. Т. 57. № 4. – с. 30-35.
11. Базлов Ю. А., Герасимов А. П., Ефремов Г. Н., Насретдинов К. К. Параметры связи систем координат // Геодезия и картография, 1996, № 8. – с. 6 – 7.

12. Балыбердин А., Синецкий В., Соловьев И. Об информационном обеспечении реализации национальной морской политики // Морской сборник. 2007. № 4. С. 25–30.

13. Баранов Ю. К. Определение места судна с помощью навигационных спутников. М.: Транспорт, 1984. 112 с.

14. Баранов Ю. К., Песков Ю. А. Теоретические основы определения судна по навигационным искусственным спутникам Земли: Учебное пособие. И.: ЦРИА «Морфлот», 1977. 106 с.

15. Богданов В. А., Рогальский В. И. О возможности создания единой спутниковой системы поиска и спасания // Экспресс информ. Сер. «Судовождение и связь», ЦБНТИ ММФ. 1982, № 4. С. 1-11.

16. Башарин В. Г. Анализ очередей в вычислительных сетях. М.: Наука, 1989. – 334 с.

17. Башарин В. Г. Модели Информационно – вычислительных систем. М.: Наука, 1993. – 69 с.

18. Башарин Г. П. Модели информационно – вычислительных систем: Сборник научных трудов. М.: Наука, 1994. – 78 с.

19. Бескид П.П., Байков Е.А., Истомин Е.П., Соколов А.Г., Фокичева А.А. Геоинформационное управление как современный подход к управлению пространственно-распределенными системами и территориями// Ученые записки российского государственного гидрометеорологического университета. 2015. №41. С. 220-239

20. Бескид П.П., Куракина Н.И., Орлова Н.В. Геоинформационные системы и технологии. СПб, РГГМУ, 2013 – 173 с.

21. Богданов В. А. Оценка составляющих ошибки определения местоположения аварийных радиобуев в спутниковой системе поиска и спасания //Тр. ЦНИИМФ. 1984. Вып. 288. С. 26-31.

22. Богданов В. А. Устранение неоднозначности навигационных определений в низкоорбитальных спутниковых радионавигационных системах//Зарубежная радиоэлектроника. 1983, № 9. С. 34-36.

23. Богданов В. А. Оценка времени передачи аварийной информации в спутниковой системе поиска и спасания КОСПАС – САРСАТ // Экспрессинформ., Сер. «Судовождение и связь», ЦБНТИ ММФ. 1983. № 5. С. 11-9.

24. Болдырев В. С., Пересыпкин В. И., Якушенков А. А. Международные стандарты точности судовождения // Морской флот. 1985. № 8. С. 28-29.

25. Брейдо В.Л. Вычислительные системы, сети и телекоммуникации/. СПб: Питер, 2003. – 688 с.

26. Буланже Ю. Д., Богданов В. И., Лазаренко Н. Н. Проблема Кронштадтского футштока //Записки по гидрографии, 1990, № 222, с. 50 – 55.

27. Бурханов М.В., Малкин И.М. Навигация с ЭКНИС. М.: Моркнига, 2014 – 298 с.

28. Бутусов М.М., Галкин С.Л. Волоконная оптика в судовом приборостроении. Л.: Судостроение, 1990.

29. Веселов, Н.В. Экспертное обеспечение транспортной логистики: Монография / Н.В. Веселов, А.А. Рогов, И.С. Кравчук, О.А. Бортник. – М.: Издательско-торговая корпорация «Дашков и К°», 2013. – 230 с.

30. Воскресенский В.В., Доценко С.М., Чудаков О.Е. Информационное обеспечение управления и флот. СПб, Ника, 2002 г.

31. Волынкин А. И., Кудрявцев И. В., Мищенко И. Н., Шебшаевич В. С. Аппаратура потребителей СРНС «Навстар», Зарубежная радиоэлектроника. 1983. № 4. С. 70-79. № 5. С. 59-66.

32. Гаскаров Д.В., Истомин Е.П., Кутузов О.И. Сетевые модели распределенных автоматизированных систем.- СПб.: Энергоатомиздат, Санкт-Петербургское отделение, 1998. - 353 с. с ил.

33. Дейт К. Введение в базы данных. –М.: Вильямс, 1999

34. Директоров Н.Ф., Катанович А.А. Современные системы внутрикорабельной связи. СПб.: Судостроение, 2001.

35. Джексон П. Введение в экспертные системы.-Москва, С. Петербург, Киев: Изд. дом “Вильямс”, 2002.-622 с.

36. Дмитриев В.И. Навигация и лоция: учебное пособие / - М.: Морреч-центр, 2015. – 360 с.

37. Дюбуа Д., Прад А. Теория возможностей. Приложения к представлению знаний. –М.: Радио и связь, 1995

38. Ездаков А.Л. Экспертные системы САПР: учебное пособие / - М.: ИД ФОРУМ, 2012. - 160 с.

39. Емельянова Н.З., Партыка Т.Л., Попов И.И.. Проектирование информационных систем: Учебное пособие / - М.: Форум: НИЦ ИНФРА-М, 2014. - 432 с.

40. Ермолаев Г. Г., Захаров В. К. Морская лоция. – М.: Транспорт, 1969. – 368 с.

41. Захаров В. К. Морская лоция. – М.: Морской транспорт, 1962. – 415 с.

42. Истомин Е.П., Слесарева Л.С., Соколов А.Г., Реализация общих законов развития организационно-технических систем в процессе геоинформационного управления// Информационные технологии и системы: управление, экономика, транспорт, право . 2013. №41. С. 114-116

43. Кашин Л. А. Построение классической астрономо-геодезической сети России и СССР(1816–1991 гг). Научно-технический и исторический обзор. – М.: Картоцентр – Геодезиздат, – 1999. – 192 с.

44. Ка-
шин Л. А. Пулковская астрономическая обсерватория и её роль в развитии астрономо-геодезии //Геодезия и картография, 1992, № 7, с 58 – 63.

45. Каяндер А.Н., Сорочинский В. А., Яшкевич Е. В. Алгоритмы определения места судна по данным доплеровских спутниковых навигационных систем//Тр. ЦНИИМФ. 1976. Вып. 216. С. 15—24.

46. Кеннеди М., Копп С. Картографические проекции. – М.: Изд-во “Дата +”, 2002. – 114.

47. Коверзнев Е.А., Сурков Д.М. Анализ надежности связи в системе КОСПАС-САРСАТ // Научный вестник Московского государственного технического университета гражданской авиации. 2006. № 99. С. 111-115.

48. Колесникова Д.Н. Моделирование систем с использованием теории массового обслуживания /Под ред. д.т.н. Учеб. Пособие /СПбГПУ. СПб, 2003. – 180 с.

49. Коломейчук Н. Д. Гидрография: Учеб. для курсантов высших военно-морских училищ. – Л.: Изд. ГУНиО МО СССР, 1988. – 363 с.

50. Корнеев В.В., Гарев А.Ф., Васюшин С.В., Райх В.В. Базы данных. Интеллектуальная обработка информации. – М.: Изд-во «Нолидж», 2000. – 351 с.

51. Кофман А. Введение в теорию нечетких множеств М.: Радио и связь, 1982. – 432с.

52. Кутузов О.И., Татарникова Т.М. Инфокоммуникационные сети. Моделирование и оценка вероятностно-временных характеристик Санкт-Петербург : ГУАП, 2015. - 381 с

53. Лукаевич И.Я. Инструментальные средства разработки экспертных систем для ПЭВМ / Человек и компьютер, №7, 2012.

54. Майоров А.А. Состояние и развитие геоинформатики // Науки о Земле. Вып. 03.2012. С. 11-16.

55. Маркелов В. М., Соловьёв И. В., Цветков В.Я. Интеллектуальные транспортные системы как инструмент управления // Государственный советник. 2014. № 3. С. 42-49.

56. Экспертные системы : учебное пособие / сост. А. Н. Никулин. – Ульяновск : УлГТУ, 2015. – 78 с.

57. Савиных В.П., Цветков В.Я. Развитие методов искусственного интеллекта в геоинформатике // Транспорт Российской Федерации. 2010. №

58. Обрезков А.И. Анализ систем массового обслуживания с неоднородными потоками заявок. – СПб.: ВУС, 1999. – 74 с.

59. Попов Э.В. Экспертные системы реального времени. В: Открытые системы, N 2 (10), 1995.

60. Пржибыл П., Свитек М. Телематика на транспорте. М.: МАДИ (ГТУ), 2003. 540 с.

61. Рыжиков Ю. И. Опыт расчёта сложных систем массового обслуживания / Ю. И. Рыжиков, А. В. Уланов // Информационно-управляющие системы. - 2009. - № 2. - С. 56-61

62. Савиных В.П., Цветков В.Я. Геоданные как системный информационный ресурс // Вестник Российской академии наук. 2014. Т. 84. № 9. С. 826-829.

63. Свиридов А.П. Нейросети и их приложения. – М: Изд-во МГСУ “Союз”, 2001. – 90 с.

64. Серапи-нас Б.Б. Введение в Глонасс и GPS измерения. Учебное пособие. – Ижевск: Удм. гос. ун-т, 1999. – 93 с.

65. Соловьёв И.В. и др. Единое информационно-управляющее пространство ВМФ. От идеи до реализации / под ред. В.И. Кидалова. СПб.: Ника, 2003. 490 с.

66. Соснин В.В. Время ожидания в неоднородных системах с очередями при обслуживании заявок в порядке поступления// Известия высших учебных заведений. Приборостроение. 2014. Т. 57. № 4. С. 39-41.

67. Справочный документ. – Под общ. ред. Дrajнюка А.А. Единая государственная система геодезических координат 1995 года (СК-95). – М.: ЦНИИГАиК, 2000. – 32 с.

68. Татарникова Т.М., Яготинцева Н.В. Имитационная модель установления соединения на сети// Научный альманах. 2016. № 2-2 (16). С. 393-397.

69. Татарникова Т.М., Яготинцева Н.В. Модель создания единого информационного пространства на корабле // Научный альманах. 2016. № 1-1 (15). С. 517-521.

70. Татарникова Т.М., Яготинцева Н.В. Модель информационной транспортной сети корабля //В сборнике: Региональная информатика и информационная безопасность Сборник трудов. СПИИРАН. 2016. С. 415-418

71. Татарникова Т.М., Яготинцева Н.В. Принципы организации экспертной системы выбора структуры локальной сети корабля //В сборнике: Современное общество, образование и наука сборник научных трудов по материалам Международной научно-практической конференции: в 16 частях. 2015. С. 142-144.

72. Татарникова Т.М., Яготинцева Н.В. Интеграция образцов радиоэлектронного вооружения на основе информационной транспортной сети корабля//В сборнике: Региональная информатика и информационная безопасность сборник трудов. Санкт-Петербургское общество информатики, вычислительной техники, систем связи и управления. 2015. С. 148-151

73. Татарникова Т.М., Яготинцева Н.В. Предоставление услуг передачи данных на корабле// В сборнике: Инновационные технологии в сервисе Сборник материалов IV Международной научно-практической конференции. Под ред. А. Е. Карлика. 2015. С. 321-322

74. Татарникова Т.М., Яготинцева Н.В. Оценка функциональной надежности корабельной сети передачи данных // Известия высших учебных заведений. Приборостроение. 2014. Т. 57. № 9. С. 62-66.

75. Татарникова Т.М., Яготинцева Н.В. Модель оценки характеристик локальной вычислительной сети корабля // В сборнике: Теоретические и прикладные проблемы науки и образования в 21 веке сборник научных трудов по материалам Международной заочной научно-практической конференции: в 10 частях. 2012. С. 143-144.

76. Татарникова Т.М., Яготинцева Н.В. Характеристика проблемы интеграции образцов радиоэлектронного вооружения корабля // Ученые записки Российского государственного гидрометеорологического университета. 2012. № 25. С. 156-162

77. Цветков В.Я. Применение геоинформационных технологий для поддержки принятия решений // Известия высших учебных заведений. Геодезия и аэрофотосъемка. 2001. № 4. С. 128-138.

78. Цветков В. Я. Информационные единицы сообщений // Фундаментальные исследования. 2007. № 12. С. 123-124.

79. Яснев, В.Н. Информационные системы и технологии в экономике : учебное пособие /-3-е изд., перераб. и доп. - М. : Юнити-Дана, 2012. - 561 с

Приложение

Имитационная модель локальной сети ГИС корабля

Листинг программы на C ++

Claim.h

```
#include "definition.h"
#ifndef CLAIM_H
#define CLAIM_H

class Claim
{
public:
    Claim();

    ~Claim( void );

    // Устанавливает время заявки
    void setCreatingTime( double time );

    // Возвращает создание времени заявки
    double getCreatingTime( void );

    // Приоритеты в очереди
    void setPriorityLevel( size_t level );

    // Определение приоритетной очереди
    size_t getPriorityLevel( void );

    // Устанавливает обработку узла коммутации(УК)
    void setStatus( bool procStatus );

    // Определяет статус узла коммутации
    bool getStatus( void );

    // Установление узла коммутации
    void setNodeType( bool newNodeType );

    // Возвращает вид узла
    bool getNodeType( void );

    // Модифицирует значение с порядковым nodeId в буфере ожидания
    void setWaitingTime( size_t nodeId, double newValue );

    // Возвращение значения заполнения буфера с nodeId
    double getWaitingTime( size_t nodeId );

    // Устанавливает обработку времени для УК
    void setProcessingTime( double newProcessingTime );

    // Возвращает время процесса в УК
    double getProcessingTime( void );

    // Показывает количество УК
    void setProcNodeNumber( size_t newProcNodeNumber );

    // Обработка УК
    size_t getProcNodeNumber( void );

    // Возвращение среднего времени системного времени
    double getTotalTimeInSystem( void );
};
#endif
```



```

//    Время ожидания по всему виртуальному каналу
double getTotalWaitingTime( void );

//    Обновления флага времени
void setProcessingState( bool procState );

//    Возвращение значений пакетов
bool getProcessingState( void );

//    Прибытие следующего пакета
void setCameOutState( bool newState );

//    Обработка следующего пакета
bool getCameOutState( void );

//    Создает буфер для ожидающего пакета
void createWaitingValueBuffer( size_t numNodes );

private:

    bool procStatus;
    bool nodeType;
    size_t procNodeNumber;
    double creatingTime;
    size_t priorityLevel;

    //    Флаг - установленная истина, если требование в настоящий
    момент является обработанным узлом
    bool processingState;

    //    Флаг указывает, есть ли требование вышел из системы
    bool cameOutFlag;

    //    Показатели деятельности УК
    double processingTime;

    vector<double> waitingBuffer;
};

#endif

```

Claim.cpp

```

#include "Claim.h"

Claim::Claim()
{
    procStatus = false;
    processingTime = 0.0;
    procNodeNumber = 0;
    processingState = false;
    cameOutFlag = false;
}

Claim::~Claim( void ) {}

//    Устанавливает создание времени требования
void Claim::setCreatingTime( double time ) { creatingTime = time; }

//    Возвращает создание времени требования
double Claim::getCreatingTime( void ) { return creatingTime; }

```

```

//    Приоритетный уровень требования
void Claim::setPriorityLevel( size_t level ) { priorityLevel = level; }

//    Получает приоритетный уровень требования
size_t Claim::getPriorityLevel( void ) { return priorityLevel; }

//    Устанавливает обработку состояния узла
void Claim::setStatus( bool newProcStatus ) { procStatus = newProcStatus; }

//    Возвращает обработку состояния узла
bool Claim::getStatus( void ) { return procStatus; }

//    Устанавливает вид узла
void Claim::setNodeType( bool newNodeType ) { nodeType = newNodeType; }

//    Возвращение типа узла
bool Claim::getNodeType( void ) { return nodeType; }

//    Модифицирует значение с порядковым nodeId в выходящем типе буферных
наборов узла
void Claim::setWaitingTime( size_t nodeId, double newValue ) { waitingBuffer[
nodeId ] = newValue; }

//    Возвращает значение из выходящего буфера с индексом nodeId
double Claim::getWaitingTime( size_t nodeId ) { return waitingBuffer[ nodeId
]; }

//    Устанавливает обработку времени для узла
void Claim::setProcessingTime( double newProcessingTime ) { processingTime =
newProcessingTime; }

//    Возвращает обработку времени узла
double Claim::getProcessingTime( void ) { return processingTime; }

//    Устанавливает обработку числа узла
void Claim::setProcNodeNumber( size_t newProcNodeNumber ) { procNodeNumber =
newProcNodeNumber; }

//    Получает обработку числа узла
size_t Claim::getProcNodeNumber( void ) { return procNodeNumber; }

//    Возвращает суммарное время, когда пакет находится в системе
double Claim::getTotalTimeInSystem( void ) { return getTotalWaitingTime() +
processingTime; }

//    Возвращает суммарное время ожидания, когда пакет ожидает в буфере системы
double Claim::getTotalWaitingTime( void )
{
    double totalTime = 0.0;

    for( size_t nodeId = 0; nodeId < waitingBuffer.size(); ++nodeId )
        totalTime += waitingBuffer[ nodeId ];

    return totalTime;
}

//    Устанавливает обновление очереди, переработка флага
void Claim::setProcessingState( bool procState ) { processingState = procState;
}

//    Возвращает флаг обрабатываемого пакета
bool Claim::getProcessingState( void ) { return processingState; }

//    Выход пакета из ВК

```

```

void Claim::setCameOutState( bool newState ) { cameOutFlag = newState; }

// Выход последнего пакета из ВК
bool Claim::getCameOutState( void ) { return cameOutFlag; }

// Создает буфер для ожидающего пакета значения
void Claim::createWaitingValueBuffer( size_t numNodes ) { waitingBuffer =
vector<double>( numNodes ); }

```

Node.h

```

#include "definition.h"
#ifndef NODE_H
#define NODE_H

class Node
{
public:
    Node();

    ~Node( void );

    // Устанавливает новое состояние узла
    void setNodeStatus( bool newNodeStatus );

    // Возвращает значение, если узел свободен
    bool getNodeStatus( void );

    // Устанавливает новый узел
    void setNodeUnlockTime( double newTime );

    // Возвращает время работы узла
    double getNodeUnlockTime( void );

    // Устанавливает индекс обработки требования
    void setIndexProcClaim( size_t newIdx );

    // Возвращает индекс обработки требования заявки
    size_t getIndexProcClaim( void );

    // Устанавливает текущее время обработки
    void setCurrentProcTime( double newProcTime );

    // возвращает текущее время обработки
    double getCurrentProcTime( void );

private:
    // Индекс обработки требования
    size_t idxProcClaim;
    // Флаг показывает доступен узел или нет
    bool nodeStatus;
    // Время, в котором узел должен быть разблокирован
    double nodeUnLockTime;

    double curProcessingTime;
};

#endif

```

Node.cpp

```

#include "Node.h"

```

```

Node:: Node ()
{
    // Флаг показывает доступен узел или нет
    nodeStatus = true;
    // Время когда узел должен быть открыт
    nodeUnlockTime = 0.0;
    // Индекс обработки требования
    idxProcClaim = 0;

    curProcessingTime = 0.0;
}

Node::~~Node( void ) {}

// Устанавливает новое состояние узла
void Node::setNodeStatus( bool newNodeStatus ) { nodeStatus = newNodeStatus; }

// Возвращает значение, свободен ли узел
bool Node::getNodeStatus( void ) { return nodeStatus; }

// Проверяет следующий узел, свободен ли он для обработки пакета
void Node::setNodeUnlockTime( double newTime ) { nodeUnlockTime = newTime; }

// Возвращает время через сколько будет свободен узел
double Node::getNodeUnlockTime( void ) { return nodeUnlockTime; }

// Устанавливает индекс обработки требования
void Node::setIndexProcClaim( size_t newIdx ) { idxProcClaim = newIdx; }

// Возвращает индекс обработки требования
size_t Node::getIndexProcClaim( void ) { return idxProcClaim; }

// Устанавливает текущее время обработки
void Node::setCurrentProcTime( double newProcTime ) { curProcessingTime =
newProcTime; }

// Возвращает текущее время обработки
double Node::getCurrentProcTime( void ) { return curProcessingTime; }

```

Definitions.h

```

#ifndef DEFINITIONS_H
#define DEFINITIONS_H

#include <cstdio>
#include <vector>
#include <iostream>
#include <iomanip>
#include <fstream>
#include <windows.h>
#include <math.h>

#endif

using namespace std;

```

Generator.h

```

#include "definition.h"
#ifndef GENERATOR_H
#define GENERATOR_H

class Generator

```

```

{
    public:

        //Равномерное распределение:  $x(i) = (x(i-1) * A + C) \% M$ ;
        explicit Generator( double A, double C, size_t M, double X0, size_t
numExp, size_t leftBound, size_t rightBound );

        //    Распределения Эрланга порядка k
        explicit Generator( double K, double L, size_t numExp );

        Generator::Generator( void );

        ~Generator( void );
        //    Генерирует массив заявок равномерным распределением
        void generateValuesUniform( void );

        //    Генерирует массив заявок распределением Эрланга порядка k
        void generateValuesErlang( void );

        //    Распределяет значения [ 0, 1 ]
        double normalization( double val );

        //    Расчет математического ожидания.
        double getMathExp( void );

        //    Расчет дисперсии
        double getVariance( void );

        //    Возвращает    последовательность    среднеквадратического
отклонения заявок
        double getStdDev( void );

        //    Возвращает шаг промежутка времени
        double getIntervalStep();

        //    Возвращает случайные значения массива
        vector<double> getRandomValues();

        //    Возвращает число интервалов (для гистограммы)
        size_t getNumberOfIntervals();

        //    Создает гистограмму плотности вероятности
        void createProbabilityDensityHistogram( void );

        //    Сортирует заявки по приоритетам
        void sortRandomValues( void );

        //    Возвращает максимальное случайное значение из заявок массива
        double getMaxRandomValue( void );

        //    Возвращет функцию корреляции
        double getCorrelation( size_t shift );

        //    Вычисляет факториал значений
        size_t factorial( size_t value );

        //    Генерирует значения в диапазоне [левая граница, правая
граница]
        void generateValuesUniformBounds( void );

    private:

        //    Параметры равномерного распределения
        double A, C, M, X0, leftBound, rightBound;

```

```

//    Параметры распределения Эрланга порядка k
double K, L;

//    Количество экспериментов
size_t numExp;
//    Массив генерируем заявок
vector<double> values;

//    Шаг гистограммы
double histStep;

//    число интервалов
size_t intervalsNum;
};

#endif

```

Generator.cpp

```

#include "Generator.h"

//     $x(i) = (x(i-1) * A + C) \% M$ ;
Generator::Generator( double A, double C, size_t M, double X0, size_t numExp,
size_t leftBound, size_t rightBound )
    : A( A ), C( C ), M( M ), X0( X0 ), numExp( numExp ), leftBound(
leftBound ), rightBound( rightBound ) {
    //    Инициализация
    intervalsNum = sqrt( static_cast<double>( numExp ) );
}

//    Распределения Эрланга порядка k
Generator::Generator( double K, double L, size_t numExp )
    : K( K ), L( L ), numExp( numExp )
{
    //    Инициализация
    intervalsNum = sqrt( static_cast<double>( numExp ) );
}

Generator::Generator( void ) {}

Generator::~Generator( void ) {}

//    Распределяет значения [ 0, 1 ]
double Generator::normalization( double val ) { return ( val - ( int )val ); }

//    Генерирует массив по равномерному распределению
void Generator::generateValuesUniform( void ) {
    values.push_back( normalization( fmod( X0 * A + C, M ) ) );

    for( size_t ind0 = 1; ind0 < numExp; ++ind0 )
        values.push_back( normalization( fmod( values[ values.size() - 1 ]
* A + C, M ) ) );
}

//    Генерирует значения в диапазоне [ левая граница, правая граница ]
void Generator::generateValuesUniformBounds( void ) {
    for( size_t ind0 = 0; ind0 < values.size(); ++ind0 )
        values[ ind0 ] = leftBound + ( rightBound - leftBound ) * values[
ind0 ];
}

//    Генерирует значения распределением Эрланга порядка k
void Generator::generateValuesErlang( void ) {

```

```

double value, uniformValue;

for( size_t ind0 = 0; ind0 < numExp; ++ind0 )
{
    value = 0.0;
    for( size_t ind1 = 0; ind1 < K; ++ind1 )
    {
        uniformValue = 0.0;

        while( uniformValue == 0 ) uniformValue = rand() /
static_cast<double>( RAND_MAX );

        value += ( - 1 / L * log( uniformValue ) );
    }
    values.push_back( value );
}

// Расчет математического ожидания для равномерного распределения
double Generator::getMathExp( void ) {
    double mathExp = 0.0;

    for ( vector<double>::iterator i = values.begin(); i != values.end(); ++i
)
        mathExp += ( *i );

    return ( mathExp /= values.size() );
}

// Возвращает значения дисперсии
double Generator::getVariance( void ) {
    double variance = 0.0;
    double mathExp = 0.0;

    mathExp = getMathExp();

    for ( vector<double>::iterator i = values.begin(); i != values.end(); ++i
)
        variance += ( ( ( *i ) - mathExp ) * ( ( *i ) - mathExp ) );

    return ( variance /= values.size() );
}

// Возвращает последовательность среднеквадратического отклонения заявок
double Generator::getStdDev( void ) { return sqrt( getVariance() ); }

// Возвращает шаг интервала
double Generator::getIntervalStep() { return histStep; }

// Возвращает случайное значения массива
vector<double> Generator::getRandomValues() { return values; }

// Возвращает число интервалов массива
size_t Generator::getNumberOfIntervals() { return intervalsNum; }

// Создает гистограмму плотност ивероятности
void Generator::createProbabilityDensityHistorgam( void ) {
    ofstream out;
    out.exceptions ( ifstream::failbit | ifstream::badbit | ifstream::eofbit
);

    // Создаем файл
    out.open( "hist_pract.csv", ofstream::out );

```

```

size_t ind0;
double shift = 0.0;
vector<size_t> hitCounter( values.size() );
vector<double> hitProbability( values.size() );

histStep = getMaxRundomValue() / intervalsNum;

double maxShift = max( intervalsNum * histStep, 1.0 );

// Пересчитайте число обращений к промежутку времени
for( ind0 = 0; shift < maxShift; shift += histStep )
{
    for ( vector<double>::iterator i = values.begin(); i !=
values.end(); ++i )
        if( ( ( *i ) > shift ) && ( *i ) < ( shift + histStep ) )
            hitCounter[ ind0 ]++;

        hitProbability[ ind0 ] = hitCounter[ ind0 ] / static_cast<double>(
values.size() );
        ++ind0;
    }

    for( ind0 = 0, shift = 0.0; ind0 < intervalsNum; ++ind0, shift +=
histStep )
        out << shift << ";" << hitCounter[ ind0 ] << ";" << hitProbability[
ind0 ] << endl;

    out.close();

    out.open( "randNumbers.csv", ofstream::out );

    for( ind0 = 0; ind0 < values.size(); ++ind0 )
        out << values[ ind0 ] << endl;

    out.close();

    // Создает файл
    out.open( "hist_teor.csv", ofstream::out );

    double expDist;

    // Теоритическое значение распределения Эрланга порядка k
    for( ind0 = 0, shift = 0.0; ind0 < intervalsNum; ++ind0, shift +=
histStep )
    {
        expDist = ( L * pow( L * shift, K - 1 ) * exp( ( - 1 ) * L * shift
) ) / factorial( K - 1 );
        out << shift << ";" << expDist << endl;
    }

    out.close();
}

// Метод факториала
size_t Generator::factorial( size_t value )
{
    if( value == 1 || value == 0 )
        return 1;
    else
        return value * factorial( value - 1 );
}

// Сортирует случайные заявки по возрастанию

```



```

void Generator::sortRandomValues( void ) {
    int ind0, ind1;
    double tmp;

    for ( ind0 = 0; ind0 < values.size(); ++ind0 ) {
        for ( ind1 = values.size() - 1; ind1 > ind0; --ind1 ) {
            if ( values[ ind1 ] < values[ ind1 - 1 ] ) {
                tmp = values[ ind1 - 1 ];
                values[ ind1 - 1 ] = values[ ind1 ];
                values[ ind1 ] = tmp;
            }
        }
    }
}

// Возвращает максимальное случайное значение из заявок
double Generator::getMaxRundomValue( void ) {
    double maxVal = 0.0;

    for ( vector<double>::iterator i = values.begin(); i != values.end(); ++i
)
        if( ( *i ) > maxVal )
            maxVal = ( *i );

    return maxVal;
}

// Корреляция между случайными заявками
double Generator::getCorrelation( size_t shift ) {
    double sum = 0;

    for( size_t ind0 = 0; ind0 < values.size() - shift; ++ind0 )
        sum += values[ ind0 ] * values[ ind0 + shift ];

    return ( 12 / ( static_cast<double>( values.size() ) - shift ) * sum - 3
);
}

```

main.cpp

```

#include "Generator.h"
#include "Claim.h"
#include "Node.h"

// Настройки программы

#define PRIORITY_ORDER
#define PRIORITY_LEVELS 3
#define NUMBER_OF_CLAIMS 100
#define NODES_NUMBER 3
#define INTENSITY_INTERNAL_CLAIMS 0.7
#define INTENSITY_EXTERNAL_CLAIMS 0.2
#define PROCESSING_INTENSITY 1.0

// Условие, чтобы остановить моделирование. Когда все заявки выходят
bool stopCondition( vector<Claim>& claims );

// Генерирует очередь требований для обработки
bool getCandidateIndexes( double time, size_t nodeId, vector<Claim>& claims,
vector<size_t>& claimsIndexes );

```

```

// Сортирует нисходящие входные массивы с соответствием приоритету
void sortClaimsIndexesWithPriority( vector<size_t>& indexes, vector<Claim>&
claims );

// Сортирующие заявки в течение времени
void sortingClaims( vector<Claim>& claims );

void main ( void )
{
    const size_t numExp = NUMBER_OF_CLAIMS;

    const size_t numNodes = NODES_NUMBER;
    const size_t numPriorityLevels = PRIORITY_LEVELS;

    vector<double> M( numNodes );
    vector<double> R( numNodes );

    const double K = 1;
    const double L = INTENSITY_INTERNAL_CLAIMS;

    // Интенсивность обработки узлов
    M[ 0 ] = M[ 1 ] = M[ 2 ] = PROCESSING_INTENSITY;
    // Интенсивность транзитных потоков
    R[ 0 ] = R[ 1 ] = R[ 2 ] = INTENSITY_EXTERNAL_CLAIMS;

    // Матрица для заявок находящихся в узле
    double I[ ] = { 1.0, 0.0,
                    1.0, 0.0 };

    // Имитационное моделирование
    cout << "Total num claims: " << NUMBER_OF_CLAIMS << endl << endl;

    Generator creatingRequestsTime( K, L, numExp );
    vector<Generator> processingTime( numNodes );
    vector<Generator> transitFlows( numNodes );
    vector<vector<double>> processingTimeRandomValues( numNodes );
    vector<vector<double>> transitFlowsTime( numNodes );
    vector<double> randomValues( numExp );
    vector<double> creatingTime( numExp );

    vector<size_t> claimsIndexes;

    creatingRequestsTime.generateValuesErlang();
    randomValues = creatingRequestsTime.getRandomValues();

    // Инициализация системного времени для транзитных и полезных потоков
    creatingTime[ 0 ] = 0.0;
    for( size_t ind0 = 1; ind0 < numExp; ++ind0 )
        creatingTime[ ind0 ] = creatingTime[ ind0 - 1 ] + randomValues[
ind0 - 1 ];

    // Инициализация время процесса
    for( size_t ind0 = 0; ind0 < numNodes; ++ind0 )
    {
        processingTime[ ind0 ] = Generator( K, M[ ind0 ], numExp );
        transitFlows[ ind0 ] = Generator( K, R[ ind0 ], numExp );

        // Генерируем случайные значения для полезного потока
        processingTime[ ind0 ].generateValuesErlang();
        processingTimeRandomValues[ ind0 ] = processingTime[ ind0
].getRandomValues();

        // Генерируем случайные значения для транзитного потока
        transitFlows[ ind0 ].generateValuesErlang();
    }
}

```

```

randomValues = transitFlows[ ind0 ].getRandomValues();

transitFlowsTime[ ind0 ].resize( numExp );
transitFlowsTime[ ind0 ][ 0 ] = 0.0;

for( size_t ind1 = 1; ind1 < numExp; ++ind1 )
    transitFlowsTime[ ind0 ][ ind1 ] = transitFlowsTime[ ind0 ][
ind1 - 1 ] + randomValues[ ind1 - 1 ];
}

// Инициализация распределения
size_t totalNumClaims = numExp * ( numNodes + 1 );
vector<Claim> claims( totalNumClaims );

// Полезный узел
size_t idx = 0;
for( size_t claimIdx = 0; claimIdx < numExp; ++claimIdx, ++idx )
{
    claims[ idx ].setCreatingTime( creatingTime[ claimIdx ] );
    claims[ idx ].setPriorityLevel( rand() % numPriorityLevels );
    claims[ idx ].setProcNodeNumber( 0 );
    claims[ idx ].setNodeType( true );
    claims[ idx ].createWaitingValueBuffer( numNodes );
}

// Оставшиеся узлы
for( size_t nodeIdx = 0; nodeIdx < numNodes; ++nodeIdx )
{
    for( size_t claimIdx = 0; claimIdx < numExp; ++claimIdx, ++idx )
    {
        claims[ idx ].setCreatingTime( transitFlowsTime[ nodeIdx ][
claimIdx ] );
        claims[ idx ].setPriorityLevel( rand() % numPriorityLevels );
        claims[ idx ].setProcNodeNumber( nodeIdx );
        claims[ idx ].setNodeType( false );
        claims[ idx ].createWaitingValueBuffer( numNodes );
    }
}

// Сортирующие заявки в течение времени
sortingClaims( claims );

// Инициализация узлов
vector<Node> nodes( numNodes );

const double timeStep = 1.0;

for( double time = 0.0; stopCondition( claims ); time += timeStep )
{
    // Проверка узла
    for( size_t nodeIdx = 0; nodeIdx < numNodes; ++nodeIdx )
    {
        if( !nodes[ nodeIdx ].getNodeStatus() )
        {
            // Узел занят
            if( time >= nodes[ nodeIdx ].getNodeUnlockTime() )
            {
                // Узел освобожден
                nodes[ nodeIdx ].setNodeUnlockTime( 0.0 );
                nodes[ nodeIdx ].setNodeStatus( true );

                claims[ nodes[ nodeIdx ].getIndexProcClaim()
].setProcessingState( false );
            }
        }
    }
}

```

```

//      Передавайте запрос в следующий узел
claims[ nodes[ nodeId ].getIndexProcClaim()
].setProcNodeNumber( nodeId + 1 );

//      Обновления узла для следующего пакета
claims[ nodes[ nodeId ].getIndexProcClaim()
].setProcessingTime(
        claims[ nodes[ nodeId
].getIndexProcClaim() ].getProcessingTime() +
        nodes[ nodeId ].getCurrentProcTime() );

claims[ nodes[ nodeId ].getIndexProcClaim()
].setCreatingTime(
        claims[ nodes[ nodeId
].getIndexProcClaim() ].getCreatingTime() +
        claims[ nodes[ nodeId
].getIndexProcClaim() ].getWaitingTime( nodeId ) +
        nodes[ nodeId ].getCurrentProcTime()
);

if( nodeId + 1 == numNodes )
{
    //      Заявка вышла из системы
claims[ nodes[ nodeId
].getIndexProcClaim() ].setStatus( true );
}

//      Проверьте условие заявки на выходе узла
if( ( rand() / static_cast<double>( RAND_MAX ) ) > I[ 2
* nodeId ] )
{
    claims[ nodes[ nodeId
].getIndexProcClaim() ].setCameOutState( true );
}
}

//      Обработка всех узлов
for( size_t nodeId = 0; nodeId < numNodes; ++nodeId )
{
    if( !getCandidateIndexes( time, nodeId, claims,
claimsIndexes ) )
        continue;

    for( size_t claimIdx = 0; claimIdx < claimsIndexes.size();
++claimIdx )
    {
        if( nodes[ nodeId ].getNodeStatus() )
        {
            double procTime = processingTimeRandomValues[
claims[ claimsIndexes[ claimIdx ] ].getProcNodeNumber() ][ rand() % numExp ];
            nodes[ nodeId ].setNodeUnlockTime( claims[
claimsIndexes[ claimIdx ] ].getCreatingTime() +
                                                    claims[
claimsIndexes[ claimIdx ] ].getWaitingTime( nodeId ) + procTime );

            nodes[ nodeId ].setNodeStatus( false );
            nodes[ nodeId ].setCurrentProcTime( procTime );
            nodes[ nodeId ].setIndexProcClaim(
claimsIndexes[ claimIdx ] );

            claims[ claimsIndexes[ claimIdx
].setProcessingState( true );
        }
        else

```

```

        {
            // Полезный узел занят? то ожидаем окончание
            полезного потока
            claims[ claimsIndexes[ claimIdx ]
].setWaitingTime( nodeId, claims[ claimsIndexes[ claimIdx ] ].getWaitingTime(
nodeIdx ) + timeStep );
        }
    }
}

/*****/
// Имитационное моделирование успешно завершено
/*****/

// Параметры результатов
double averageClaimsWaitingTime = 0.0;
size_t counter = 0;

for( size_t claimIdx = 0; claimIdx < totalNumClaims; ++claimIdx )
{
    if( claims[ claimIdx ].getNodeType() == true )
    {
        averageClaimsWaitingTime += claims[ claimIdx
].getTotalWaitingTime();
        ++counter;
    }
}
cout << "\nThe average time waiting -->INTERNAL<-- claims in system is:
";

averageClaimsWaitingTime / counter << endl;

averageClaimsWaitingTime = 0.0; counter = 0;

for( size_t claimIdx = 0; claimIdx < totalNumClaims; ++claimIdx )
{
    if( claims[ claimIdx ].getNodeType() == false )
    {
        averageClaimsWaitingTime += claims[ claimIdx
].getTotalWaitingTime();
        ++counter;
    }
}
cout << "The average time waiting -->EXTERNAL<-- claims in system is: ";

averageClaimsWaitingTime / counter << endl;

double averageClaimsSpentTime = 0.0;
counter = 0;

for( size_t claimIdx = 0; claimIdx < totalNumClaims; ++claimIdx )
{
    if( claims[ claimIdx ].getNodeType() == true )
    {
        averageClaimsSpentTime += claims[ claimIdx
].getTotalTimeInSystem();
        ++counter;
    }
}
cout << "\nThe average time spent -->INTERNAL<-- claims in system is: ";

averageClaimsSpentTime / counter << endl;

```

```

averageClaimsSpentTime = 0.0; counter = 0;

for( size_t claimIdx = 0; claimIdx < totalNumClaims; ++claimIdx )
{
    if( claims[ claimIdx ].getNodeType() == false )
    {
        averageClaimsSpentTime += claims[ claimIdx
].getTotalTimeInSystem();
        ++counter;
    }
}
cout << "The average time spent -->EXTERNAL<-- claims in system is: ";

cout << averageClaimsSpentTime / counter << endl;

cout <<endl;

vector<size_t> deletedClaims( numNodes, 0 );
vector<size_t> restClaimsAfterNode( numNodes, 0 );
for( size_t nodeIdX = 0; nodeIdX < numNodes - 1; ++nodeIdx )
{
    for( size_t claimIdx = 0; claimIdx < totalNumClaims; ++claimIdx )
    {
        if( ( claims[ claimIdx ].getProcNodeNumber() - 1 ) == nodeIdX
&&
            claims[ claimIdx ].getCameOutState() == true )
            ++deletedClaims[ nodeIdX ];
    }

    if( nodeIdX == 0 )
    {
        cout << "Removed claims after node " << nodeIdX << " is: " <<
deletedClaims[ nodeIdX ] << " / "
            << numExp * 2 << " <-- ";

        cout << "[" << I[ 2 * nodeIdX + 1 ] << " ," << I[ 2 * nodeIdX
] << "]" ( matrix row flipped! )" << endl;

        restClaimsAfterNode[ nodeIdX ] = numExp * 2 - deletedClaims[
nodeIdx ];
    }
    else
    {
        cout << "Removed claims after node " << nodeIdX << " is: " <<
deletedClaims[ nodeIdX ] << " / "
            << numExp + restClaimsAfterNode[ nodeIdX - 1 ] << " <-
- ";

        cout << "[" << I[ 2 * nodeIdX + 1 ] << " ," << I[ 2 * nodeIdX
] << "]" ( matrix row flipped! )" << endl;

        restClaimsAfterNode[ nodeIdX ] = numExp +
restClaimsAfterNode[ nodeIdX - 1 ] - deletedClaims[ nodeIdX ];
    }
}

cout << "\nDone!" << endl;
}
// Условие, чтобы остановить моделирование. Когда все заявки выходят
bool stopCondition( vector<Claim>& claims )
{
    size_t counterUnprocClaims = 0;

    for( size_t claimIdx = 0; claimIdx < claims.size(); ++claimIdx )

```

```

    {
        if( claims[ claimIdx ].getNodeType() )
        {
            if( claims[ claimIdx ].getStatus() == false && claims[
claimIdx ].getCameOutState() == false )
                ++counterUnprocClaims;
        }
    }
    cout << "The number of unprocessed claims: " << counterUnprocClaims <<
endl;

    if( counterUnprocClaims != 0 ) return true;
    else return false;
}

// Генерирует очередь требований для обработки
bool getCandidateIndexes( double time, size_t nodeId, vector<Claim>& claims,
vector<size_t>& claimsIndexes )
{
    // Выполняется очистка массива
    claimsIndexes.resize( 0 );
    for( size_t claimIdx = 0; claimIdx < claims.size(); ++claimIdx )
    {
        if( claims[ claimIdx ].getProcNodeNumber() == nodeId &&
            !claims[ claimIdx ].getStatus() &&
            !claims[ claimIdx ].getProcessingState() &&
            !claims[ claimIdx ].getCameOutState() &&
            claims[ claimIdx ].getCreatingTime() < time )

            claimsIndexes.push_back( claimIdx );
    }

#ifdef PRIORITY_ORDER
    sortClaimsIndexesWithPriority( claimsIndexes, claims );
#endif

    if( claimsIndexes.size() > 0 ) return true;
    else return false;
}

// Сортирует нисходящие входные массивы с соответствием приоритетов
void sortClaimsIndexesWithPriority( vector<size_t>& indexes, vector<Claim>&
claims )
{
    for ( size_t ind0 = 0; ind0 < indexes.size(); ++ind0 )
    {
        for ( size_t ind1 = indexes.size() - 1; ind1 > ind0; --ind1 )
        {
            if ( claims[ indexes[ ind1 ] ].getPriorityLevel() < claims[
indexes[ ind1 - 1 ] ].getPriorityLevel() )
            {
                size_t temporary = indexes[ ind1 - 1 ];
                indexes[ ind1 - 1 ] = indexes[ ind1 ];
                indexes[ ind1 ] = temporary;
            }
        }
    }
}

// Сортирует заявки в течение времени
void sortingClaims( vector<Claim>& claims )
{
    for ( size_t ind0 = 0; ind0 < claims.size(); ++ind0 )
    {

```

```
for ( size_t ind1 = claims.size() - 1; ind1 > ind0; --ind1 )
{
    if( claims[ ind1 ].getCreatingTime() < claims[ ind1 - 1
].getCreatingTime() )
    {
        Claim temporary = claims[ ind1 - 1 ];
        claims[ ind1 - 1 ] = claims[ ind1 ];
        claims[ ind1 ] = temporary;
    }
}
}
```