

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ
ВЫСШЕГО ПРОФЕССИОНАЛЬНОГО ОБРАЗОВАНИЯ «РОССИЙСКИЙ
ГОСУДАРСТВЕННЫЙ ГИДРОМЕТЕОРОЛОГИЧЕСКИЙ УНИВЕРСИТЕТ»

На правах рукописи

Колбина Ольга Николаевна

Разработка геоинформационной системы оценки параметров климатических условий на основе
распределенных гетерогенных баз данных

Специальность 25.00.35 – Геоинформатика

Диссертация
на соискание ученой степени
кандидата технических наук

Научный руководитель:

доктор технических наук,
профессор Е.П. Истомин

Санкт-Петербург – 2014

ОГЛАВЛЕНИЕ

ВВЕДЕНИЕ.....	- 3 -
1. Модели и методы распределенной обработки геоданных	- 10 -
1.1 Обоснование и выбор параметров геомоделирования в арктической зоне РФ.....	- 10 -
1.2. Современные информационные технологий в области обработки информации - 21 -	
1.3 Распределенные ГИС	- 36 -
1.4 Технологии построения информационных систем на основе гетерогенных баз данных	- 48 -
Вывод по главе.	- 56 -
2. Разработка модели геоинформационной системы на основе гетерогенных баз данных	- 58 -
2.1 Модели информационно-управляющих систем.....	- 58 -
2.1.1 Общие понятия моделирования	- 58 -
2.1.2 Модель информационной системы управления рисками нагонных наводнений в городе Санкт-Петербург	- 63 -
2.2 Разработка модели геоинформационной системы на основе гетерогенных баз данных	- 68 -
2.3 Модификация методики проектирования геоинформационной системы на основе распределенных гетерогенных баз данных	- 80 -
Вывод по главе.	- 89 -
3. Информационная система обработки геоданных в Арктическом региона РФ	- 91 -
3.1. Математическая модель оценки исследуемых геоданных	- 91 -
3.2. Моделирование геоинформационной системы «Оценка климатических параметров для ведения хозяйственной деятельности на арктической территории РФ».....	- 104 -
3.3 Методика оценки с использованием ГИС.....	- 114 -
Вывод по главе.	- 120 -
ЗАКЛЮЧЕНИЕ	- 122 -
Список сокращений и условных обозначений	- 124 -
СПИСОК ЛИТЕРАТУРЫ.....	- 125 -
Приложение А. Скрипт «блока подготовки данных» принятых от системы SOLab.....	- 130 -
Приложение Б. Свидетельство о регистрации электронного ресурса «Программа расчета георисков и вероятности появления нагонного наводнения».....	- 149 -
Приложение В. Свидетельство о регистрации базы данных «Ice_Concentration».....	- 150 -
Приложение Г. Свидетельство о регистрации программы для ЭВМ «Прогноз нефтяного разлива»..	- 151 -

ВВЕДЕНИЕ

Актуальность исследования. В современном мире северные регионы России и Арктика занимают всё более значимые позиции среди остальных территорий РФ. Это связано, во-первых, с их ролью в формировании глобального климата и устойчивости состояния биосферы; во-вторых, наличием обширных запасов углеводородов на их территории и подводных месторождений Арктики; в-третьих, военно-стратегическим и транспортным потенциалом региона.

Одним из факторов оказывающим влияние на социально-экономическое развитие Арктической зоны являются экстремальные природно-климатические условия, включая низкие температуры воздуха, сильные ветры и наличие ледяного покрова на акватории арктических морей.

Для ведения любой хозяйственной деятельности в регионе будь то ведение разведки и добычи углеводородного сырья, или судоходство, или ведение рыболовного промысла необходима точная и своевременная информация о климатических условиях на период проведения работ, наличие которой обеспечивает безопасную и продуктивную работу.

Данные относительно высоты волн, скорости ветра, океанических течений, температуры играют важную роль при выборе методов перемещения судов и персонала с платформы на берег, выборе типов оборудования, необходимого для проведения запланированных работ, выборе конструкции эксплуатационных платформ, а также объектов инфраструктуры, способной выдержать установленный срок службы, и выборе стратегий ликвидации чрезвычайных ситуаций и многое другое.

Внедрение информационных технологий в деятельность организаций привело к развитию распределенных информационных систем (РИС), реализующих различные информационные среды на основе разнообразных баз данных. Это привело к появлению неоднородного информационного пространства, реализующего гетерогенные базы данных. К управлению такими распределенными системами предъявляются повышенные требования с точки зрения обеспечения целостности и непротиворечивости данных, степени однородности информационного пространства. Проблема состоит в технологической сложности перехода от централизованного однородного управления данными к распределенной обработке данных в гетерогенной информационной среде, включающей автономные системы работы в локальной сети гетерогенных баз данных, не представляющие возможность контроля выполнения глобальных транзакций. Принципы создания распределенных геоинформационных систем на

основе гетерогенных баз данных, в том числе, могут быть применены для успешного освоения арктических территорий РФ.

На сегодняшний день можно говорить о недостаточном развитии методов хранения и использования геоинформации в распределенных гетерогенных базах данных.

Вследствие чего подтверждается актуальность проблемы совершенствования методов хранения и использования геоинформационных данных в различных форматах и их пространственно обособленных системах, базах данных, решению которой посвящена диссертация. Методы включают в себя оперативное извлечение и обработку данных, которые хранятся в различных системах и форматах, с целью своевременной обработки и предоставления информации о состоянии характеристик окружающей среды, влияющих на ведение хозяйственной деятельности.

Степень разработанности проблемы. В процессе исследования были изучены труды отечественных и зарубежных ученых по рассматриваемой проблеме, таких как Фуфаев Э.В., Оззу М.Т., Валдуриз П., Гаскаров Д.В., Кутузов О.И., Диго С.М., Б.П. Арсеньев, С.А.Яковлев., Бурковский В.Л., Дорофеев А.Н., Семенин С.В., Акальцов В.П., Ивакин А.Я., Шкодырев В.П., Середович В.А. и других.

Проанализированы материалы научных организаций, нормативно-правовые документы, программы и концепции федеральных и региональных органов власти, связанные с развитием арктического региона и проблемами информационного обеспечения его освоения.

Проблематика хранения и использования геоданных в различных форматах и их пространственно обособленных системах и базах данных в России изучена недостаточно. Не в полной мере разработаны методы и технологии хранения и использования геоинформации на основе распределенных, гетерогенных баз данных и знаний. Таким образом, необходимость совершенствования механизмов работы с распределенными геоданными различных форматов определяет объект и предмет, цель и задачи диссертации.

Объектом исследования являются геоинформационные системы обработки пространственно обособленных данных различных форматов.

Предметом исследования выступает разработка технологий использования гетерогенных баз данных в распределенных географических информационных системах.

Цель диссертационной работы заключается в совершенствовании технологий хранения и использования геоинформации на основе распределенных гетерогенных баз данных.

Реализация поставленной цели исследования предопределяет постановку следующих **задач:**

1. Обосновать требования к геоданным используемым в прикладных геоинформационных системах с элементами управления;
2. Построить модель геоинформационной системы с учетом препроцессорной подготовки гетерогенных данных;
3. Модифицировать методику проектирования информационно-управляющей системы с учетом распределенных гетерогенных баз данных;
4. Разработать математическую модель обработки гетерогенных данных в рамках верификации геоинформационной системы.

Теоретические и методические основы исследования. Теоретической основой исследования являются труды отечественных и зарубежных ученых в области проблематики хранения и использования геоданных в различных форматах, распределенных информационных систем, аналитические обзоры, посвященные анализу технологий доступа к различным видам данных, нормативно-правовые документы, связанные с методикой построения информационных систем. Методической основой исследования является:

- Обобщение и анализ существующего опыта создания распределенных информационных систем с гетерогенными базами данных;
- Системный анализ и концептуальное моделирование;
- Аналитические исследования;
- Математическое моделирование;
- Прогнозирование;
- Метод геоинформационного моделирования.

Обоснованность и достоверность результатов исследования, выводов и рекомендаций обеспечивается:

– внутренней непротиворечивостью результатов исследования и их соответствием теоретическим положениям фундаментальных исследований в области технологий сбора, регистрации, хранения, передачи и обработки геоинформации с использованием вычислительной техники, телекоммуникационных систем распространения пространственно-временной геоинформации, технологий хранения и использования геоинформации на основе распределенных баз данных и знаний;

– использованием для достижения цели работы нормативных документов, программ, документов федеральных и региональных органов власти, касающихся методики построения информационных систем;

– применением принципов системного анализа и концептуального моделирование, аналитических исследований, математического моделирования и других современных научных методов;

– апробацией результатов исследования на научно-практических конференциях и отражением основных результатов диссертации в открытой печати.

При решении поставленных в работе задач получены следующие **результаты, выносимые на защиту:**

1. Требования к геоданным, используемым в прикладных геоинформационных системах с элементами управления;
2. Модель геоинформационной системы с препроцессорной подготовкой гетерогенных данных;
3. Модифицированная методика проектирования информационно-управляющей системы на основе распределенных гетерогенных баз данных;
4. Математическая модель обработки гетерогенных данных в рамках верификации геоинформационной системы.

Научная новизна состоит в том, что:

- были обоснованы требования к геоданным используемым в геоинформационных системах с элементами управления, что позволит сократить сроки подготовки данных;
- впервые представлена модель геоинформационной системы на основе гетерогенных распределенных данных, охватывающая все слои представления информации: данные, обработка, интерфейс. Разработаны принципы и средства, позволяющие эффективно размещать и синхронизировать доступ к пространственным геоданным в различных форматах без их конвертации. Для объединения и преобразования данных к желаемому для пользователя виду используется «виртуальный процессор данных». Предложенная модель позволяет сократить время доступа к разноформатным геоданным;
- в методике впервые объединены технологии создания информационно-управляющей и геоинформационной системы, осложненной гетерогенностью и распределенностью баз данных, входящих в её основу. В связи, с чем предложено создание виртуальной интегрированной базы данных, а также описание информационных ресурсов системы в виде реляционной базы данных. В методике усовершенствована технология, позволяющая привлекать разноформатные данные;
- математическая модель предложенная для верификации ГИС впервые применяется в исследуемых системах отгашенных гетерогенными БД. В геоинформационной системе проводится анализ не только поступающих локальных данных, но и привлекаются из удаленного сервера данные, преобразующиеся с помощью «блока подготовки данных» в необходимый формат для их последующего использования. Полученные таким образом данные средствами проверки статистических гипотез и алгоритмов, позволяющих решать задачи прогнозирования множества состояний при полной определенности исходных значений, могут

быть использованы, в том числе, и для прогнозирования выбранных климатических параметров. Представленная геоинформационная система существенно отличается от имеющихся тем, что в основе моделирования лежат не физические, а математические вероятностные модели.

Теоретическая значимость результатов диссертации заключается в том, что в работе был обобщен опыт практического применения технологий сбора и обработки информации, телекоммуникационных систем распространения пространственно-временной геоинформации, построения распределенных и гетерогенных баз данных, научно-методического аппарата создания информационных систем. Предложены оригинальные концептуальные принципы взаимодействия разноформатных источников информации, баз данных с прямым пользователем, и на их основе разработаны новые структурные модели геоинформационных систем на основе гетерогенных баз данных и усовершенствована методика создания информационно-управляющей системы с учетом распределенных гетерогенных баз данных.

Практическая и научная значимость работы заключается в том, что решена научно-техническая задача, имеющая существенное значение для геомоделирования и системного анализа многоуровневой и разнородной информации: исследованы технологии сбора и обработки информации с использованием вычислительной техники, телекоммуникационных систем распространения пространственно-временной геоинформации, построение распределенных и гетерогенных баз данных, возможность совершенствования технологий хранения и использования геоинформации на основе распределенных гетерогенных баз данных по средствам применения новых методик и моделей построения ГИС, что соответствует п.п. 7, 9 области исследования паспорта специальности 25.00.35 «Геоинформатика».

Предложенные в работе оригинальные авторские разработки могут быть использованы на практике в реальных условиях создания специализированных геоинформационных систем, использующих данные, находящиеся в распределенных и различных по форматам хранения базах данных. Для этого в работе даются конкретные рекомендации по их разработке, выделены основные этапы проектирования, построены диаграммы адекватности модели. Основные выводы и положения диссертационного исследования были использованы в научно-исследовательских работах и учебном процессе.

Апробация работы:

1. Свидетельство о регистрации электронного ресурса «Программа расчета георисков и вероятности появления нагонного наводнения» №19308. 24.09.2013 г.
2. Свидетельство о регистрации базы данных «Ice_Concentration» №2014621110, дата регистрации 07.08.2014 года;

3. Победитель конкурса грантов для студентов вузов, расположенных на территории Санкт-Петербурга, аспирантов вузов, отраслевых и академических институтов, расположенных на территории Санкт-Петербурга в 2013 году

4. Международная научно-практическая конференция «Инфогео 2013» секция «геоинформационные системы» (26-28 ноября 2013г.)

5. Международная научно-практическая конференция «Инфогео 2014» секция «геоинформационные системы» (3-6 октября 2014г.)

6. Участие в НИР №1223 «Разработка и развитие методов, моделей и систем геоинформационного управления пространственно-распределенными объектами», 2013 год.

7. Участие в НИР №74.20.56 «Разработка методических основ геоинформационного управления рисками развития рекреационных приморских территорий», 2014 год.

Структура и объем диссертации. Диссертация состоит из введения, трех глав, заключения, списка использованной литературы, списка сокращений и условных обозначений, приложений. Объем работы составляет 150 листов, в том числе 50 рисунков, 4 таблицы и три приложения. Библиография диссертационной работы составляет 67 наименований.

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

В **первой главе** «Модели и методы распределенной обработки геоданных» приводятся результаты анализа современных технологий доступа к данным, заключающихся в определении нехватки простого метода работы с разноформатными данными. Сформулированы требования к геоданным, используемым в прикладных геоинформационных системах с элементами управления, рассмотрены основные направления и существующие проблемы в их реализации. Отобраны параметры геомодели прогнозирования состояния климатических условий в арктической зоне РФ.

Во **второй главе** «Разработка модели геоинформационной системы на основе гетерогенных баз данных» представлена разработанная модель геоинформационной системы с препроцессорной подготовкой гетерогенных данных. Показаны современные проблемы создания геоинформационной системы, осложненной гетерогенностью и распределенностью баз данных, и определена модель системы способная усовершенствовать работу по использованию информации, учитывающая разноформатность и распределенность

предоставленных данных. Модифицирована методика проектирования информационно-управляющей системы на основе распределенных гетерогенных баз данных.

В *третьей главе* «Информационная система обработки геоданных в Арктическом регионе РФ» представлена математическая модель обработки гетерогенных данных в рамках верификации геоинформационной системы, которая позволяет одновременно использовать данные для оценок вероятностных характеристик двух параметров (силы ветра и плотности льда). Приводятся результаты моделирования геоинформационной системы «Оценка климатических параметров для ведения хозяйственной деятельности на арктической территории РФ», подтверждающие возможность реализации выбранной модели геосистемы. Разработана первичная Геоинформационная система «Оценка климатических параметров для ведения хозяйственной деятельности на арктической территории РФ».

В *заключении* сформулированы выводы и предложения, вытекающие из результатов исследования, определены направления дальнейших исследований.

1. Модели и методы распределенной обработки геоданных

Работая с информацией, человеку доступен весь мир. Тенденции развития информационных технологий указывают на массовость и доступность информации. Принимается всё больше правительственных решений о необходимости информатизации общества. Начиная с трехглавого закона «Об информации, информатизации и защите информации» принятого ещё в 1995 году и заканчивая федеральным законом «Об электронной подписи» принятым 2011 годом. [11] Развитие техники и технологий подталкивает человечество к новой эре – эре информационного общества, где информация доступна любому человеку, в любое время и в любом месте. Это требует создание таких информационных систем, которые способны предоставить информацию из множества различных источников, не зависимо от их места расположения. Данные технологии уже существуют, но применяются в основном для создания специализированных информационных систем в крупных корпорациях. Применение же распределенной обработки для геоданных встречается не так часто как при создании распределенных автоматизированных систем. Хотя геоинформационные системы предлагают больше возможностей и наглядности пользователю.

1.1 Обоснование и выбор параметров геомоделирования в арктической зоне РФ

Полярной территорией Российской Федерации называют территорию, находящуюся за границей северного полярного круга (Рисунок 1.1.1).



Рисунок 1.1.1 Границы Северного полярного круга

В русском языке область от Северного полярного круга до Северного полюса называется «Заполярье» или «Арктикой».

Не стоит путать понятия Северного полюса и Арктики.

Северный полюс — это точка, в которой ось вращения Земли пересекает её поверхность в Северном полушарии, располагается в центральной части Северного Ледовитого океана. [33]

Согласно международному праву, Северный полюс и прилегающий к нему регион Северного Ледовитого океана в настоящее время не принадлежат ни одной стране.

Арктика — это район Земли, примыкающий к Северному полюсу и включающий окраины материков Евразии и Северной Америки, почти весь Северный Ледовитый океан с островами (кроме прибрежных островов Норвегии), а также прилегающие части Атлантического и Тихого океанов. Площадь Арктики — около 27 млн км². Если Арктику ограничить с юга Северным полярным кругом (66° 33' с. ш.), в этом случае её площадь составит 21 млн км² (Рисунок 1.1.2). [33]

Арктика поделена на пять секторов ответственности между США, Россией, Норвегией, Канадой и Данией. Тем не менее, точная граница Арктики не определена.

Между тем Россия имеет свою Арктическую зону. В соответствии с законом от 2 мая 2014 года "О сухопутных территориях Арктической зоны Российской Федерации" в её состав входят [34]:

1. Мурманскую область;
2. Ненецкий автономный округ;
3. Чукотский автономный округ;
4. Ямало-Ненецкий автономный округ;
5. Муниципальное образование городского округа "Воркута" (Республика Коми);
6. Территории Аллаиховского улуса, Анабарского национального (Долгано-Эвенкийского) улуса, Булунского улуса, Нижнеколымского района, Усть-Янского улуса (Республика Саха (Якутия));
7. Территории городского округа города Норильска, Таймырского (Долгано-Ненецкого) муниципального района, Туруханского района (Красноярский Край);
8. Территории муниципальных образований "город Архангельск", "Мезенский муниципальный район", "Новая Земля", "Город Новодвинск", "Онежский муниципальный район", "Приморский муниципальный район", "Северодвинск" (Архангельская область);
9. Земли и острова, расположенные в Северном Ледовитом океане, некоторые улусы Якутии. Эти географические объекты были объявлены территорией Советского Союза Постановлением Президиума ЦИК СССР от 15 апреля 1926 года.



Рисунок 1.1.2 Арктический регион

Север России омывают Белое море, Баренцево, Карское, Восточно-Сибирское и Чукотское, Берингово. На Севере несколько тысяч озер и сотни рек. Крупнейшие озера — Ладожское и Онежское, реки — Северная Двина, Печора, Мезень, Онега, Обь, Енисей, Лена, Колыма. Северные моря, озера и реки богаты рыбой. Большая часть территории лежит в зоне вечной мерзлоты. Продолжительная зима сменяется коротким и, как правило, прохладным

летом. Земли за Северным Полярным кругом (широта 66°33'), составляющие более 20% территории России.

На Севере полностью или частично расположены 27 (из 89) субъектов Российской Федерации. Построены города, порты, горные и промышленные центры: **Мурманск**, Норильск, Воркута, Салехард, Нарьян-Мар, Дудинка, Игарка, Диксон, Тикси и др. Здесь проживает около 10 млн. человек, что составляет приблизительно 7% населения страны, из них около 180 тыс. человек относятся к малочисленным народам Севера.

Активное освоение Севера началось в XVII в., когда русские землепроходцы пришли в Заполярье. В XVIII в. Великая Северная экспедиция (Х.П. и Д.Я. Лаптевы, С.Г. Малыгин, С.И. Челюскин др.) обследовала и нанесла на карту почти все северное побережье Азии. По проекту М.В. Ломоносова в 1765–1766 гг. была снаряжена первая русская высокоширотная экспедиция под начальством В.Я. Чичагова. Исследования Заполярья были продолжены в XIX — начале XX в. Огромную роль в освоении Заполярья сыграл **Северный морской путь**. Он связал европейские и дальневосточные порты России, а также устья судоходных рек **Сибири**, что способствовало освоению природных богатств этих районов и их экономическому развитию. Впервые Северный морской путь был пройден за одну навигацию экспедицией О.Ю. Шмидта на судне «Сибиряков» в 1932 г. [32]

Традиционными видами хозяйственной деятельности коренных народов Севера является пушной промысел и оленеводство.

Арктика богата природными ресурсами. Здесь сосредоточено до 80% запасов минерально-сырьевых ресурсов страны, свыше 60% лесных богатств и 90% запасов пресной воды.

В настоящее время активизация работ по разведке и освоению ископаемых в Арктике является стратегическим интересом не только России, но и других государств. Особое внимание уделяется на арктическом шельфе. Российские нефтегазовые корпорации заключают крупные контракты с зарубежными компаниями на разработку месторождений углеводородов.

Дальнейшее ведение деятельности по добыче ископаемых в Арктике станет стимулом развития смежных отраслей экономики. Необходимость транспортировки углеводородов даст толчок созданию современной инфраструктуры и модернизации ледокольного флота. Организация работ по разведке новых месторождений гарантирует создание новых рабочих мест, в том числе, для жителей приарктических территорий. Указанные факторы создают предпосылки успешного социально-экономического развития северных территорий страны.

В то же время активное освоение месторождений ископаемых создаёт угрозу экосистеме региона.

Рисунок 1.1.3 Северный Морской путь

Альтернатива же Северному морскому пути — транспортные артерии, проходящие через Суэцкий или Панамский каналы. Однако если, например, расстояние, проходимое судами из порта Мурманск в порт Иокогаму (Япония) через Суэцкий канал, составляет 12840 морских миль, то Северным морским путём — только 5770 морских миль (Рисунок 1.4.4).



Рисунок 1.1.4 Альтернатива Северному морскому пути - путь через Суэцкий канал

Развитие Арктической зоны напрямую связано с развитием навигационно-гидрографического и гидрометеорологического обеспечения мореплавания, что сказывается на уровне освоения территорий и экономических рисках такого освоения. Оптимальным решением развития территории региона является создание организационно-технических систем на основе ГИС, способных предоставлять актуальную информацию по состоянию и прогнозированию климатических явлений органам государственного и коммерческого управления.

Специфика современного управления развитием сложных, в том числе и пространственно-распределенных организационно-технических систем (ОТС), способных оказать влияние на развитие целого региона, определяется рядом факторов (рисунок 1.1.5) [15].

Процесс глобализации представляет собой движение потоков материи, энергии и информации в геопространственных масштабах — в той части Вселенной, где существует антропосфера — область деятельности людей. Для управления развитием пространственных ОТС фактор глобализации предполагает необходимость рассмотрения перспектив развития

ОТС и управленческих решений как элемента глобальной (мировой) социально-экономической системы.

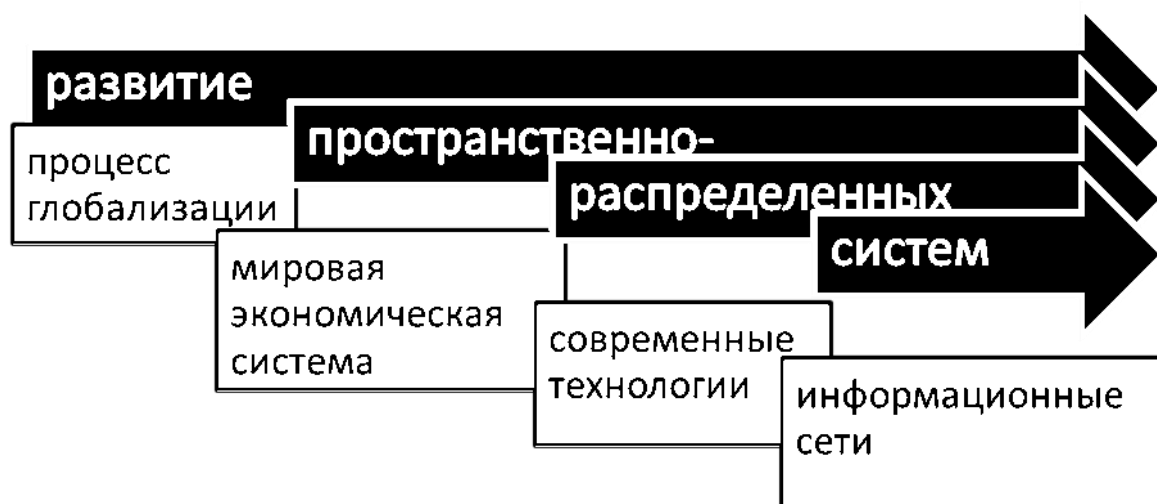


Рисунок 1.1.5 – Основные факторы развития организационно-технических систем

Мировая социально-экономическая система включает мировой рынок капиталов, продукции, человеческих ресурсов и др. Кроме того, существенное влияние на управление развитием пространственных ОТС оказывают объективные факторы – уровень обеспеченности ресурсами, природные условия, географическое положение и т.п.

Уровень развития современных технологий позволяет не ограничивать пространственные масштабы управления хозяйственной деятельностью территорией отдельной страны (региона), субъекта управления.

Наличие и возможности информационных сетей различного масштаба от локальных до глобальных позволяют реализовать в управлении развитием пространственных ОТС принципы сетевой экономики.

Поэтому современные факторные условия позволяют по-новому взглянуть на систему управления как сложную, распределенную в геопространстве систему взаимодействия различных субъектов и объектов управления – что и объясняет возникновение феномена геоинформационного управления.

Определим геоинформационное управление, как управление с привлечением пространственной информации. Пространство при этом следует рассматривать в широком смысле – как логически мыслимую форму, которая служит средой существования других форм и тех или иных конструкций, объектов. Объекты как конструкции пространства представляют собой точки, характеризующиеся совокупностью свойств в соответствии с определенными для пространства отношениями.

В этом смысле геоинформационное управление организационно-техническими системами представляет собой управление, основанное на привлечении информации об объектах, существующих в геопространстве.

Феномен геоинформационного управления в современных условиях приводит к необходимости рассматривать и реализовать систему управления деятельностью человека как многомерное пространственное явление, включающее определенным образом распределенные объективные и субъективные факторы, и реализуемое в геопространстве. [15]

С точки зрения геоинформационного управления геопространство можно структурировать и выделить требования к представлению геоинформации (Рисунок 1.1.6.).



Рисунок 1.1.6 – Структура пространства решений в геоинформационном управлении

Координатное (географическое) пространство предполагает размещение объектов, конструкций в принятой системе координат, например, трехмерное пространство - x, y, z – ϕ, λ, h , или другая принятая система координат.

Координатное пространство характеризуется:

- координатами объектов;
- расстояниями,
- геометрическими фигурами,
- границами и др.

Факторное пространство – размещение в географическом пространстве объективных и субъективных факторов, влияющих на принятие и реализацию управленческих решений.

Характеризуется:

- координатами размещения – например, координаты положения в трехмерном пространстве - φ, λ, h ;
- особенностями и параметрами факторов - природные, ресурсные, предпринимательские и др.;
- доступностью и затратами для их использования в целях управления и др.

Пространство событий. Событие представимо как ограниченное во времени и пространстве явление, конкретный временной факт, результат деятельности системы, которые свершаясь, отменяют или дополняют прежние результаты наблюдения и исследования системы. События индивидуализируются относительно системы в своей уникальной, неповторимой и ограниченной во времени и пространстве сущности. С точки зрения геоинформационного управления событие — то, что имеет место, происходит, наступает в произвольной точке пространства и времени, существенное значимое происшествие, явление или иная деятельность как факт существования системы, оказывающее влияние на управленческие решения.

Событие как элемент научного познания имеет широкий спектр толкований:

- природное явление (геологическое, физическое, биологическое, экологическое, космологическое и т.п.);
- историческое, психобиографическое событие («история жизни»);
- мировое событие (катастрофы, войны, эпидемии);
- событие в статусе происшествия или случая (событийность повседневного опыта).

Событие есть явление, обретшее индивидуальную выраженность, даже собственное имя. В этом смысле событийной формой наделяются все научные открытия, получающие имена ученых, впервые их открывших, получают имена различные стихийные явления и аномалии, исторические эпохи и политические события. Осуществляясь, событие вносит изменения в область собственного осуществления и тем самым оказывает влияние на апостериорный и априорный анализ процессов развития организационно-технических систем.

Событие характеризуется:

- координатами (φ, λ, h),
- временем возникновения (t),
- масштабами (пространственными, временными),
- последствиями (уровнем воздействия на пространственную ОТС) и др.

Пространство состояний объектов управления (фазовое или атрибутивное пространство) относится к конкретному объекту управления. Динамика существования объектов во времени (жизненный цикл существования) и в пространстве ограничивает и изменяет условия управления и, соответственно, управленческие решения. Использование

пространства состояний для управления развитием объектов позволяет определить **требования** к созданию современной ГИС, заключающиеся в необходимости:

- исследовать связи субъектов и объектов управления для различных временных срезов геоинформационного управления;
- сформировать многомерные динамические модели принятия решений на различных уровнях геоинформационного управления;
- создавать динамические (адаптивные) модели управления потоками информации (базами данных и знаний) в процессе управления.

Методы исследования и модели подобных систем в пространстве состояний должны быть ориентированы на исследование процесса развития конкретного объекта, динамики изменения его состояний (атрибутов) во времени и пространстве, как под влиянием внутренней среды, так и под влиянием системы отношений с другими объектами и внешними условиями существования. Такой подход позволит использовать динамическую атрибутивную логику геоинформационного управления объектами различных масштабов.

Пространство состояний объектов управления характеризуется некоторой совокупностью переменных (атрибутов). Поэтому каждое такое состояние можно представить вектором n -мерного пространства переменных состояния, включающим:

- координаты размещения объектов в географическом пространстве (φ, λ, h) ;
- взаимное влияние объектов - система отношений, связей;
- свойства, атрибуты, в том числе время;
- значимость для управления, например, вес для выбора альтернатив;
- степень участия в событиях и др.

Если вектор представляет собой совокупность чисел, такое пространство называют фазовым пространством, а фигура (линия), которую описывает конец вектора во времени – называется траекторией движения или развития объекта в пространстве состояний. В таком представлении можно говорить об отображении множества объектов на множество векторов, а отношения между объектами пространства сводятся к отношениям на множестве векторов.

Объекты могут характеризоваться не только числами, но и другими свойствами, которые не являются числами (форма, цвет, материал и др.). Совокупность таких объектов также можно рассматривать как кодированное векторное пространство. При этом свойства кодируются с помощью чисел или каких-либо символов, которые можно истолковать как составляющие векторов (объектов) пространства. Подобные коды используют для передачи сообщений, обработки информации и т.п. В простейших случаях кодирование состояния исследуемого объекта может быть сведено к простой нумерации – объект рассматривается как номер (адрес) в базе данных, в которой хранится информация о присущих ему свойствах.

Можно говорить об объектах как точках пространства, не связывая эти объекты с обычным представлением о векторах, как последовательностях чисел, кодов или символов. При таком подходе пространство состояний можно рассматривать как пространство объектов – формализованное с помощью некоторых принятых описаний (слова, понятия, люди, животные, механизмы, организации и т.п.). Операции на множестве таких объектов выполняются по специально устанавливаемым правилам, алгоритмам, а отношения между ними выражаются в соответствующей вербальной, математической, логической или другой форме (словесные, символические, образные и другие отношения и связи).

Пространство решений составляет основу геоинформационного управления. Управленческое решение представляет собой волевой акт выбора оптимального (в смысле принятого критерия) варианта решения из перечня заранее разработанных альтернатив. Решение объединяет все элементы геопространства и характеризуется:

координатами границ области реализации решения – зона ответственности, время разработки и реализации решения, например - $\varphi, \lambda, h, \Delta t$;

уровнем решения - юрисдикция, подотчетность объектов управления, масштаб;

степенью формализации - нормы, стандарты, законы, обязательность исполнения;

содержанием - исполнители, время реализации, технологии реализации, результаты и форма их представления и др. [15]

В пространстве решений существенную роль играют субъективные факторы – культура, динамизм населения, лидерство, существующие и потенциальные возможности (развитость науки, экономики, технологий и др.).

Существующие уровни пространства решений:

надстрановой - глобальный, международный, макрорегиональный;

страновой, микрорегиональный (регионы государства);

уровень местного самоуправления (муниципальный);

организационный;

групповой, индивидуальный.

Особенность пространства решений – уровни принятия решений, в определенном смысле (с некоторыми ограничениями), образуют иерархическую систему – связаны между собой системой вертикальных связей. Сложность современной пространственной ОТС и множество субъектов управления элементами со своими целями приводит к ограниченности иерархии управления и возрастанию значимости системы горизонтальных связей.

1.2. Современные информационные технологий в области обработки информации

Человечество работает с информацией с того времени как к человеку пришло осознание что он человек.

Работа с информацией подразумевает её хранение, передачу и обработку. Начиная с 70-х годов 20 века, манипуляция с данными основывается на вычислительной технике, что привело к созданию большого количества ориентированных на определённую область деятельности информационных систем. Ядро любой информационной системы составляет информационная база, чаще всего в роли которой выступает база данных.

База данных как одно из направлений теории информации представляет собой методы и средства разработки компьютерных информационных систем, основу которых составляют особым образом структурированные файлы, предоставляющие пользователю эффективные методы получения и анализа данных, необходимых для принятия оптимального решения. [1]

Базы данных представляют собой особую организацию данных и предполагают использование специальных программно-языковых средств – систем управления базами данных (СУБД). [2]

В соответствии с ГОСТ определение базы данных звучит так: совокупность взаимосвязанных данных, организованных в соответствии со схемой базы данных таким образом, чтобы с ними мог работать пользователь. [4]

До того, как была разработана первая СУБД, доступ к данным осуществлялся прикладными программами на прямую, а сами данные были организованы в виде плоских файлов. Возникавшие при этом проблемы с логической целостностью данных, а также невозможность представить логические связи между ними в таких системах послужили причиной создания первой модели данных – иерархической.

Иерархическая модель данных допускает только два вида связей между сущностями: «один-к-одному» и «один – ко - многим». При реализации связи «многие – ко – многим» необходимо создание дополнительных таблиц их дублирование. Иерархические базы данных по сути своей являются навигационными. Это означает, что доступ возможен только с помощью заранее определенных связей и должен следовать иерархии, то есть производиться, начиная с её вершины. В иерархии прямой доступ к данным крайне ограничен, а обычно и вовсе не возможен. Достоинства такой базы данных в том, что её навигационная природа обеспечивает очень быстрый доступ при следовании вдоль заранее определенных связей. Но

негибкость модели, и невозможность наличия у сущности нескольких родителей делают её непригодной в условиях частого выполнения запросов, не запланированных заранее. [3]

Развитие информационных технологий подвигло разработчиков баз данных на создание двух моделей, описывающих сети связей между данными. Их назвали простая и сложная сетевые модели. Они должны были устранить ограничения, которые были существенным недостатком в иерархической модели.

В простой сетевой модели все связи имеют тип «один-к-одному» или «один – ко – многим». Связь «многие – ко – многим» не допускается, но разрешена множественная подчиненность, что решает проблемы избыточного дублирования и ограничений на виды доступа. Прямой доступ к экземплярам сущностей поддерживается за счет хеширования, но, поскольку последнее влияет на физическое размещение данных в файле, на практике быстрый доступ возможен лишь к одной сущности в иерархии.

Простые сетевые базы данных демонстрируют хорошую производительность, если доступ осуществляется с помощью заранее определенных связей, но работают медленно, когда в программе приходится кодировать поиск, не основанный, ни на таких связях, ни на хешированных ключах. Таким образом удовлетворить незапланированные запросы может оказаться практически невыполнимой задачей.

Сложная сетевая модель аналогична простой, но допускает реализацию связей типа «многие – ко – многим». Однако если со связью ассоциированы некоторые данные, то и тут приходится формировать композиционную сущность. Коммерчески успешных реализаций сложных сетевых СУБД так и не появилось, в основном из-за того, что связи «многие – ко – многим» стали чересчур запутанными и сложными для сопровождения. [3]

При внимательном просмотре эволюций моделей, можно увидеть, что все они похожи между собой, но создание реляционной модели данных приобрело статус прорыва в мировой индустрии информационных технологий.

Реляционная модель принципиально отличается от всех ранее существующих тем, что связи между таблицами не являются составной частью базы. Логические связи представлены только соответствиями между первичными и внешними ключами, которые при необходимости используются для выполнения операции соединения. Поэтому реляционные базы данных не являются навигационными и предоставляют поддержку незапланированных запросов.

Традиционно недостатком реляционных баз данных считалась сравнительно низкая производительность. Иерархическая и сетевая модели тесно связаны с физическим размещением информации, а связи являются неотъемлемой частью базы, но доступ к связанным данным удастся выполнить гораздо быстрее. Однако при реализации данного

подхода на современных быстродействующих компьютерах, низкая производительность уже не является непреодолимым препятствием.

Появление **объектно-ориентированного подхода** внесло глобальные изменения во взгляды на данные и процедуры их обработки. Традиционно информация и процедуры хранились отдельно, данные и связи между ними в базе данных, а процедуры – в прикладной программе. А объектная модель позволяет хранить процедуры обработки сущностей вместе с данными. Сущности становятся замкнутыми единицами, которые сравнительно легко можно использовать повторно и перемещать в новое место. Таким образом, поведение сущности не привязано к конкретной прикладной программе, а является частью самой сущности, так как вне зависимости от того, где сущность используется, она ведет себя предсказуемым, заранее известным образом. Формулирование незапланированных запросов к такой базе данных осуществляется не так просто, как к реляционной, но возможность их поддержки существует. [3]

К объектно-ориентированным базам данных можно отнести такие БД, как IBM Lotus Notes/Domino, Jasmin, ObjectStore, Cache и другие. [9]

На сегодняшний день повсеместно применяют и работают с большим количеством разнообразных баз данных, которые распределяются по следующим понятиям:

1. настольные;
2. многопользовательские;
3. удаленные;
4. распределенные.

Их, в свою очередь, можно сгруппировать по двум признакам:

1. однопользовательские (настольные);
2. многопользовательские (удаленные).

Под удаленными следует понимать такие базы данных, доступ к информации и управление которыми осуществляются с помощью линий связи. [1]

Распределённая база данных представляет собой распределённые данные и связи между ними.

Под распределенными данными понимаются физически или даже географически разделенные данные по разным компьютерам одной сети. [5]

В современных информационных технологиях БД обычно регистрируются в системе драйверов, где указывают псевдоним или лучше сказать, имя системного ресурса. Он играет огромную роль в связи логического идентификатора БД и её физического построения.

По указанному псевдониму можно узнать физическое размещение БД на жёстком диске, тип используемой СУБД, параметры страничной организации и многое другое. То есть через

псевдоним можно идентифицировать любую БД и подключиться к ней. Так, к примеру, в реляционной модели БД псевдоним указывает на имена таблиц, столбцов и атрибуты. (Рисунок 1.2.1)



Рисунок 1.2.1 Структурная схема распределённой базы данных с использованием псевдонима

Следовательно, распределенные базы данных можно определить, как совокупная среда распределённых данных при существовании в этой среде данных хотя бы двух элементов данных с разными псевдонимами. [5]

С точки зрения пользователей удаленная база данных выглядит как обычная настольная база данных, компоненты которой могут находиться на различных узлах локальной сети организации. [1] Невозможно сегодня представить информационную систему без удаленного доступа к информации. Среди разработчиков программного обеспечения данная технология является стандартом.

Рассмотрим технологии доступа к данным по различным критериям (Таблица 1.2.1).

Критерий независимости от языка программирования подразумевает языки, поддерживающие методы объектно-ориентированного программирования.

Таблица 1.2.1 Анализ технологий удаленного доступа к базам данных

Критерии	Технологии доступа к данным									
	COM (Component Object Model)	MTS (Microsoft Transaction Server)	OLE DB (Object Linking and Embedding Data Base)	ADO (ActiveX Data Objects)	MIDAS (Multitier Distributed Applications Services)	CORBA (Common Object Request Broker Architecture)	DAO (Data Access Objects)	RDO (Remote Data Objects)	ODBC (Open Database Connectivity)	JDBC (Java Data Base Connectivity)
Принцип действия	Обеспечивает взаимодействие программ любого типа, где одна часть ПО использует службы ПО (клиентская), а другая обеспечивает доступ к этим службам (серверная). В основе технологии лежат COM–компоненты.	Создание многозвенных приложений. Реализация технологии распределенных вычислений.	В технологии OLE DB используется механизм провайдеров, под которыми понимают поставщиков данных. Провайдер представляет собой компонент COM, позволяющий принимать вызовы OLE DB и выполнять все необходимое для обработки запроса к источнику данных.	Обеспечивает единый доступ к различным источникам данных. Используют ADO-компоненты и OLE-технологии.	Создание многозвенных распределённых приложений.	Разработка и эксплуатация распределённых приложений. Использует брокер запросов и сетевого агента для взаимодействия с другими объектами через интерфейсы.	В основе и DAO и RDO лежит открытый интерфейс связи с базами данных (OpenDatabaseConnectivity, ODBC), который представляет собой независимую от типа баз данных технологию для организации взаимодействия с реляционными СУБД.	Представляет собой стандартный интерфейс(API) для получения и отправки источникам данных различных типов. Это набор абстрактных классов, представляющих их логические сущности в базе данных. Для доступа используются специальные драйвера.	Суть его в том, что он предоставляет программисту удобный интерфейс к абстрактному API – X/Open SQL CLI. Выполняются SQL-запросы к базам данных из программ, написанных на платформенно-независимом языке Java.	

Критерии	COM (Component Object Model)	MTS (Microsoft Transaction Server)	OLE DB (Object Linking and Embedding Data Base)	ADO (ActiveX Data Objects)	MIDAS (Multitier Distributed Applications Services)	CORBA (Common Object Request Broker Architecture)	DAO (Data Access Objects)	RDO (Remote Data Objects)	ODBC (Open Database Connectivity)	JDBC (Java Data Base Connectivity)
Предок	-	COM	COM	OLE DB	-	-	ODBC	ODBC	-	-
Независимость от операционной платформы	Да, но закрепился в основном на операционных системах семейства Microsoft Windows	Да	OC Windows	OC Windows	Да	Да	Да	Да	OC Windows	Да
Независимость от языка	Да	Да	Да	Да	Да	Да	Да	Да	Да	Да
Разработчик	Microsoft	Microsoft	Microsoft	Microsoft	Borland (Inprise)	Консорциум OMG – Object Management Group	Microsoft		Microsoft, в сотрудничестве с Simba Technologies	Sun Microsystems

Критерии	COM (<i>Component Object Model</i>)	MTS (<i>Microsoft Transaction Server</i>)	OLE DB (<i>Object Linking and Embedding Data Base</i>)	ADO (<i>ActiveX Data Objects</i>)	MIDAS (<i>Multitier Distributed Applications Services</i>)	CORBA (<i>Common Object Request Broker Architecture</i>)	DAO (<i>Data Access Objects</i>)	RDO (<i>Remote Data Objects</i>)	ODBC (<i>Open Database Connectivity</i>)	JDBC (<i>Java Data Base Connectivity</i>)
Работа с транзакциями	Да	Обеспечивает автоматическое отслеживание и изоляцию транзакций в многопользовательском режиме.	Да	Да	Да	Да	Да	Да	Да	Да
Одновременная работа нескольких пользователей	-	Одновременная работа большого количества клиентов	Да	Да	Да	Одновременная работа большого количества клиентов	Да	Да	Да	Да

Критерии	COM <i>(ComponentObjectModel)</i>	MTS <i>(Microsoft TransactionServer)</i>	OLE DB <i>(Object Linking and Embedding Data Base)</i>	ADO <i>(ActiveXDataObjects)</i>	MIDAS <i>(Multitier Distributed Applications Services)</i>	CORBA <i>(Common Object Request Broker Architecture)</i>	DAO <i>(Data Access Objects)</i>	RDO <i>(Remote Data Objects)</i>	ODBC <i>(Open Database Connectivity)</i>
Предоставляю т доступ к таблицам разных форматов	-	-	ДА	ДА	-	-	Только реляционные СУБД.	Только реляционные СУБД.	Доступ к SQL-совместимым базам данных.
Распространение платное\бесплатно е	При покупке ОС Windiws.	При подключении пакетаWindiwsNTOption Pack.	При покупке ОС Windiws.	Приобретение ПО, лицензий.	Приобретение ПО, лицензий.	Приобретен ие ПО, лицензий.	Приобретение ПО, лицензий.	Приобретение ПО, лицензий.	Приобретение ПО, лицензий.

Критерии	COM (Component Object Model)	MTS (Microsoft Transaction Server)	OLE DB (Object Linking and Embedding Data Base)	ADO (ActiveX Data Objects)	MIDAS (Multitier Distributed Applications Services)	CORBA (Common Object Request Broker Architecture)	DAO (Data Access Objects)	RDO (Remote Data Objects)	ODBC (Open Database Connectivity)	JDBC (Java Data Base Connectivity)
Достоинства	1.COM более законченная система	1. можно реализовать с высокой задержкой ресурсы как асинхронных потоков ресурсов	1.Скорость и возможность использования специфических возможностей БД (например bulk-copyrowsets в SQL server)	1. удобный прикладной интерфейс 2. существенно меньше объектов, и он проще в обращении 3. не требует значительных объемов памяти и поэтому эта модель может использоваться в Web-приложениях	Позволяет: 1. получить доступ к данным физическим на разных компьютерах 2. распределить нагрузку сети, чтобы уменьшить сетевой трафик 3. автоматическим формировать бизнес-логику и размещать в наиболее защищённых местах			Обеспечивает возможность работы с большими количествами баз данных различных разработчиков.	1.приложение может получать доступ к совершенно разным базам данных не меняя при этом код 2. простота разработки приложения; 3. технология ODBC позволяет создавать распределенные гетерогенные приложения без учета конкретных СУБД.	1.Лёгкость разработки: разработчик может не знать специфики базы данных, с которой работает; 2.Код не меняется, если компания переходит на другую базу данных; 3.Не нужно устанавливать громоздкую клиентскую программу; 4.К любой базе можно подсоединиться через легко описываемый URL.

Критерии	COM (<i>Component Object Model</i>)	MTS (<i>Microsoft Transaction Server</i>)	OLE DB (<i>Object Linking and Embedding Data Base</i>)	ADO (<i>ActiveX Data Objects</i>)	MIDAS (<i>Multitier Distributed Applications Services</i>)	CORBA (<i>Common Object Request Broker Architecture</i>)	DAO (<i>Data Access Objects</i>)	RDO (<i>Remote Data Objects</i>)	ODBC (<i>Open Database Connectivity</i>)	JDBC (<i>Java Data Base Connectivity</i>)
Недостатки	Находится на более низком уровне и при существенно большем количестве ограничений, определяемых самой концепцией системы.		Технология OLE DB является тяжеловесной, сложной и очень чувствительной к ошибкам.	Низкая производительность и ограниченные возможности управления			Лучше всего подходит для относительно небольших, локальных баз данных.	Доступ к локальным базам данных Jet и ISAM выполняется в RDO через драйверы ODBC, что снижает его быстродействие по сравнению с DAO	1. снижение скорости доступа к данным, что связано с необходимостью трансляции запросов; 2. увеличение время обработки запросов, что связано с введением доп. прог-ого слоя; 3. необходимы предварительная инсталляция и настройка ODBC-драйвера на каждом рабочем месте. 4. предоставляет доступ только к реляционным БД.	1. Производительность транзакций может быть медленнее, чем у других типов драйверов. 2. Драйвер ODBC должен быть установлен на клиентской машине. 3. Не подходит для апплетов, потому что драйвер ODBC должен быть установлен на клиенте.

Критерии	COM (<i>Component Object Model</i>)	MTS (<i>Microsoft Transaction Server</i>)	OLE DB (<i>Object Linking and Embedding Data Base</i>)	ADO (<i>ActiveX Data Objects</i>)	MIDAS (<i>Multitier Distributed Applications Services</i>)	CORBA (<i>Common Object Request Broker Architecture</i>)	DAO (<i>Data Access Objects</i>)	RDO (<i>Remote Data Objects</i>)
Используется в ГИС	ESRI ArcMap	-	STAR GIS/Viewer	-	-	ГИС ObjectLand, SAGA GIS	MapInfo Professional	gvSIG

Существует несколько технологий удаленного доступа к данным.

Каждая технология имеет свои преимущества и недостатки, каждая ориентирована на решение определенных проблем.

Какие-то технологии лучше работают с транзакциями, при этом обеспечивают автоматическое разрешение конфликтных ситуаций, какие-то могут обеспечить одновременную работу нескольких миллионов пользователей, какие-то просты в использовании и могут предоставить доступ к таблицам разных форматов.

Все рассматриваемые технологии основаны на трёх принципах:

1. Независимость от физического размещения объекта. Объекты программного приложения могут располагаться на разных компьютерах и выполняться в различных процессах.
2. Независимость от операционной платформы. Различные части одного программного приложения могут быть реализованы на разных операционных платформах.
3. Независимость от языка программирования. Не имеет никакого значения на каком языке программирования написано каждое из объединяемых приложений. Главное, чтобы эти языки были объектно-ориентированными и поддерживали базовые соглашения по обмену данными.

Начало всех разработок положила технология COM, когда корпорация Microsoft, в 1993 году разработала свой стандарт COM. На его основе были созданы такие технологии как DCOM (главный конкурент CORBA), COM+, .NET, MTS, OPC, OLE, ActiveX, ActiveCOM (рисунок 1.2.2). Некоторые из них рассмотрены в таблице 1.2.1, а другие не вошли в неё по причине идентичности показателей проводимого анализа.

Технология COM может обеспечить взаимодействие программ любого типа, где одна часть программного обеспечения использует службы (клиентская), а другая – обеспечивает доступ к этим службам (серверная). Если стандарты COM были соблюдены, то нет никакой разницы на каком языке был написан программный продукт.

Объект COM - это некоторый функционал, наделенный свойствами и методами, которые группируются в один или несколько интерфейсов, имеющих уникальные имена. Интерфейс, в свою очередь, содержит набор методов, которые выполняют обработку данных. Объект COM характеризуется своим классом, в котором реализованы коллекции интерфейсов.

В данной технологии реализуются следующие принципы:

1. Наследование реализации, когда потомок получает от предка все программные коды.

2. Наследование интерфейса, когда родители передают потомку только интерфейсы, а программный код потомок получает лишь на время и после выполнения уничтожается.

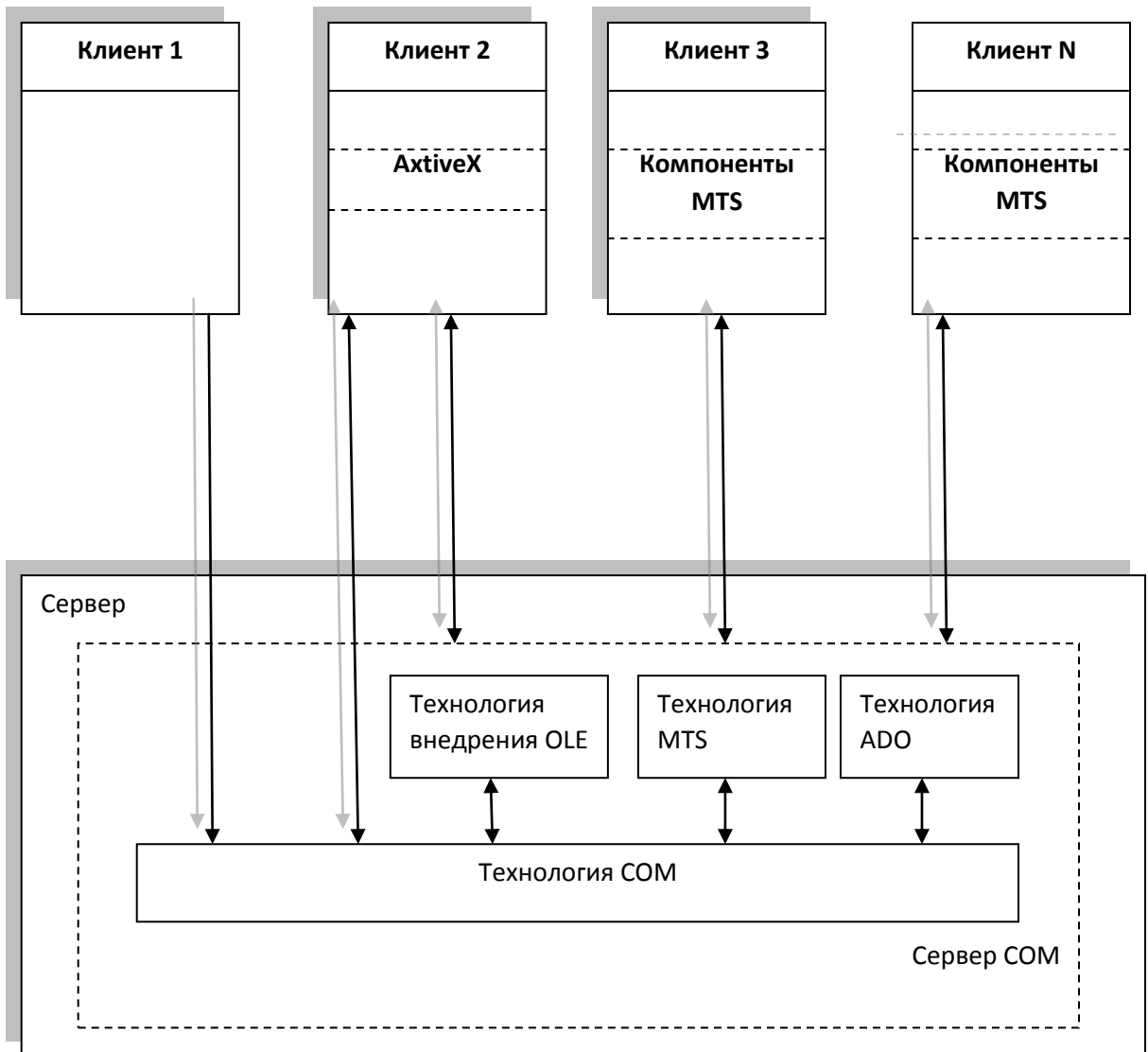


Рисунок 1.2.2 Работа клиентов в технологии COM

Технология MTS ориентирована на создание многозвенных приложений и должна обеспечивать автоматическое отслеживание и изоляцию транзакций в условиях одновременной работы большого количества клиентов.

Технология ADO обеспечивает единый доступ к различным источникам данных, что даёт возможность создания приложения клиента по управлению распределенными и разноформатными базами данных. Запросы в данной технологии формируются по одинаковым правилам и отсылаются к любому источнику данных через ADO – компоненты. Серверы баз данных напрямую выполняют эти запросы. В ней также предусмотрены механизмы доступа к последовательностям файлов, электронным таблицам и другим источникам данных. Данными

здесь могут выступать как общепринятые наборы данных, так и графические данные, которые можно использовать при создании географических информационных систем.

Технология OLE DB выступила основой для создания более усовершенствованной модели доступа к данным ADO, т.к. OLE сложна и тяжеловесна для понимания и скоростного создания приложений.

Технология MIDAS является набором сервисов для создания многозвенных распределенных приложений, которая позволяет:

1. получить доступ к данным физически расположенным на разных компьютерах;
2. распределить нагрузку сети, чтобы уменьшить сетевой трафик;
3. автоматически формировать бизнес-логику и размещать её в наиболее защищенных местах. [9]

Технология MIDAS была разработана именно для передачи информации базы данных в современных распределенных приложениях. Многозвенность данной технологии заключается в способе создания пакета данных, и посылка его приложению. Может быть использованы протоколы TCP/IP, или созданы двоичные программные компоненты и показаны приложению через интерфейсы, используя DCOM или CORBA, или могут быть преобразованы пакеты данных в документы HTML, используя ISAPI/NSAPI расширения веб-серверов. [10]

Технология CORBA предназначена для разработки и эксплуатации распределенных приложений и поддерживается большинством известных операционных систем. По своей модели обращения к базе данных она сходна с технологией COM. В ней, так же, как и в COM объект взаимодействует с другими объектами через интерфейсы. Взаимодействие происходит за счёт использования брокера запросов ORB (Object Request Broker) и сетевого агента SmartAgent, которые и позволяют работать клиентам с разными операционными платформами (Рисунок 1.2.3).

Брокер запросов ORB, по своей сути, является объектной шиной, через которую происходит соединение локальных и удаленных объектов.

Главной отличительной особенностью технологии CORBA от технологии COM можно назвать наследование объектом CORBA методов от многочисленных предков.

Экземпляр брокера запросов ORB должен быть установлен в каждом клиентском приложении. Клиентское приложение обращается к экземпляру объекта CORBA через соответствующие механизмы брокера запросов. Получив запрос, брокер запросов просматривает сетевое окружение и находит свободный работающий сетевой агент. Затем устанавливается сеанс связи с базой данных и среди зарегистрированных серверов CORBA находит свободный. Сетевой агент передаёт запрос пользователя найденному серверу CORBA

через собственный брокер. После исполнения запроса полученный ответ передаётся пользователю в обратном направлении. [9]

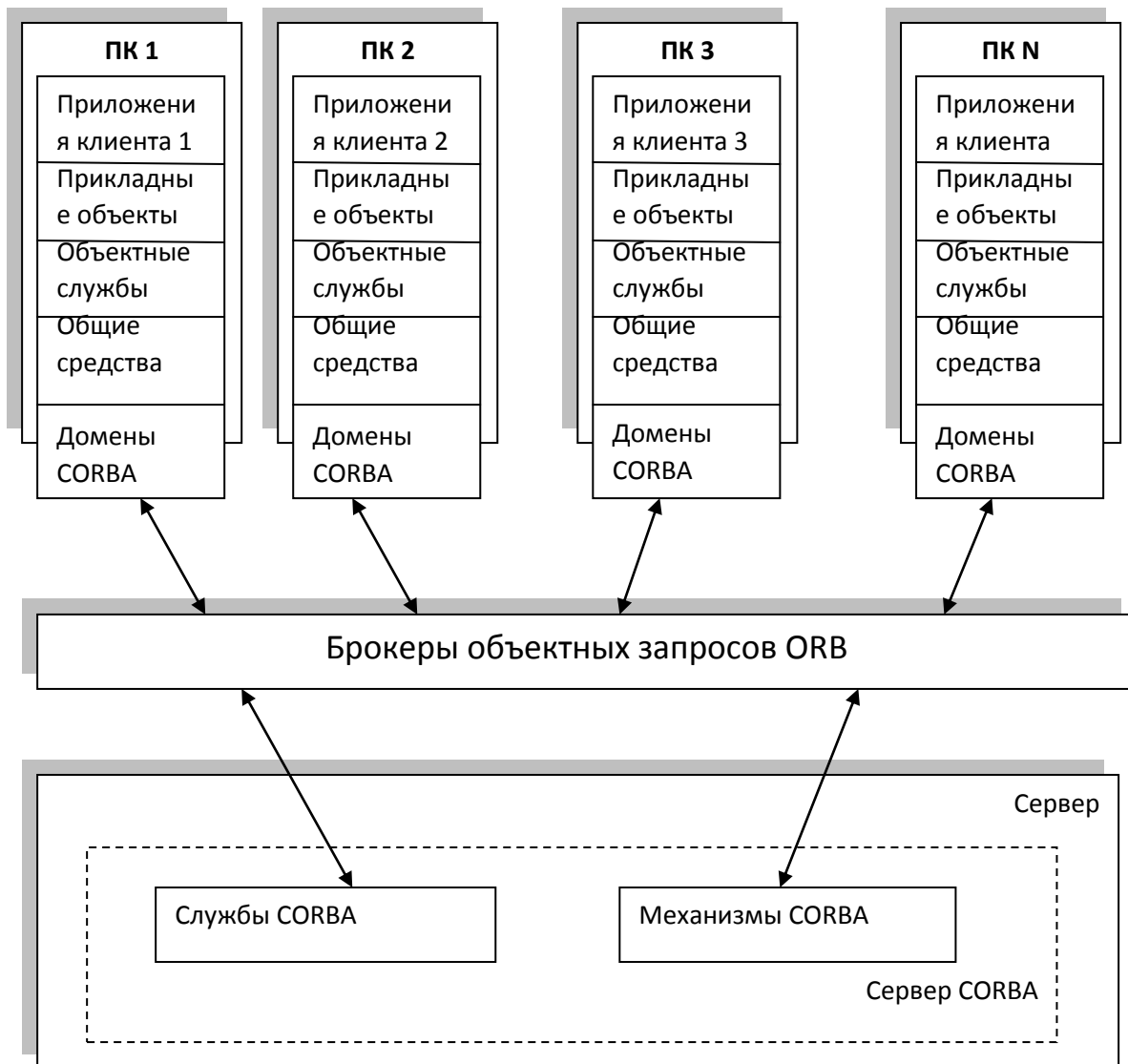


Рисунок 1.2.3 Взаимодействие сервера CORBA с клиентами

Все рассмотренные технологии непосредственно связаны друг с другом и проистекают в следствии создания другой, какие-то достоинства были отмечены и повторены во вновь созданной технологии, а недостатки удалены. Таким образом, можно наблюдать такое разнообразие доступов к удаленным данным в информационных системах. Единственным лишь ограничением в них является использование методов объектно-ориентированного программирования, что нельзя назвать недостатком, а лишь направлением применения данных технологий.

1.3 Распределенные ГИС

Компьютерные системы претерпевают революцию. С 1945 года, когда началась эпоха современных компьютеров, до 1985 года компьютеры были большими и дорогими. Ни одна крупная организация не имела больше пяти компьютеров, и, так как методы их соединения тогда были ещё не изобретены, они работали независимо друг от друга. Но скорость роста, наблюдавшаяся в компьютерных технологиях в последние полвека, действительно потрясает.

Уже в середине восьмидесятых ситуация начала меняться под воздействием двух технологических новинок. Первой можно считать разработку мощных микропроцессоров. Изначально они были 8-битными, затем стали доступны 16-, 32- и 64-битные процессоры.

Второй из новинок было изобретение высокоскоростных компьютерных сетей. Локальные сети (Local-Area Networks, LAN) соединяют сотни компьютеров, находящихся в здании, таким образом, что машины в состоянии обмениваться небольшими порциями информации за несколько микросекунд. Глобальные сети (Wide-Area Networks, WAN) позволяют миллионам машин во всем мире обмениваться информацией со скоростями, варьирующимися от 64 кбит/с до гигабит в секунду.

Благодаря развитию этих технологий сегодня не просто возможно, но и достаточно легко можно собрать компьютерную систему, состоящую из множества компьютеров, соединенных скоростной сетью. Она и называется распределенной системой (distributed system), в отличие от предшествовавших ей централизованных или однопроцессорных систем, состоявших из одного компьютера, его периферии, и, возможно, нескольких удаленных терминалов.

В литературе встречается множество различных определений распределенных систем.

По ГОСТу 34.321-96, распределенные системы определяются как информационные системы, объекты данных и/или процессы которой физически распределяются на две или более компьютерные системы. [13]

Но наиболее удовлетворяющее всем аспектам распределения определение такой системы звучит следующим образом: Распределенная система – это набор независимых компьютеров, представляющей их пользователям единой объединенной системой.

В этом определении оговариваются два момента. Первый относится к аппаратуре: все машины автономны. Второй касается программного обеспечения: пользователи думают, что имеют дело с единой системой.

Первая из таких характеристик состоит в том, что от пользователей скрыты различия между компьютерами и способами связи между ними. Другой важной характеристикой распределенных систем является способ, при помощи которого пользователи и приложения единообразно работают в распределенных системах, независимо от того, где и когда

происходит их взаимодействие. Следующей является характеристика, относящаяся к возможности легкого расширения или масштабирования системы. Данные характеристики перекликаются с двенадцатью свойствами или качествами идеальной распределенной базы данных из определения С.Д. Date, что не удивительно, так как распределенная система может включать в себя распределенную базу данных. [15]

Распределенные системы обычно существуют постоянно, однако некоторые их части могут временно выходить из строя. Пользователи и приложения не должны уведомляться о том, что эти части заменены или починены или, что добавлены новые части для поддержки дополнительных пользователей или приложений.

Для того чтобы поддержать представление различных компьютеров и сетей в виде единой системы, организация распределенных систем часто включает в себя дополнительный уровень программного обеспечения, находящийся между верхним уровнем, на котором находятся пользователи и приложения, и нижнем уровнем, состоящим из операционных систем обычно называемый системой промежуточного уровня (Рисунок 1.3.1)



Рисунок 1.3.1 Организация распределенной системы с промежуточным уровнем

Большинство организованных распределенных систем работают с обычным набором форматов данных. Это могут быть числовые, текстовые, графические данные, хранящиеся в единой базе данных или же в распределенной. Строятся они в основном по принципу самой распространенной модели организации данных – реляционной.

И действительно реляционная модель данных удобна и функциональна. Но существуют данные, которые подлежат особой организации хранения и представления – это пространственные данные.

Пространственная информация, т.е. информация об объектах, помещенных в такую пространственную систему координат, как поверхность Земли, на протяжении длительного времени воспринималась как источник проблем при организации вычислений. Все данные, что участвуют в информационных процессах, непрерывно накапливаются многочисленными

датчиками и устройствами сбора данных. К примеру, система Earth Observing System (EOS), принадлежащая НАСА, ежедневно генерирует один терабайт данных.

Снимки с искусственных спутников – яркий пример пространственных данных. Чтобы прочитать информацию с такого снимка, нужно обработать данные с учетом пространственной системы координат, которой, возможно является поверхность Земли.

Ещё в 1972 году понятием обработки пространственных данных пользовались для обозначения деятельности небольшого сообщества, члены которого занимались электронной обработкой данных для повышения производительности при составлении и редактировании карт, картографических измерений и анализе пространственных данных. Сегодня нет, наверное, той области деятельности, где бы не применялись информационные системы, работающие с пространственной информацией, называемые географической информационной системой (ГИС, англ. GIS, Geographic Information Systems). Но самое важное, что разработчики различных СУБД пришли к единогласному выводу, что пространственные или географические или геопро пространственные данные должны обрабатываться иначе, чем данные иного рода.

Пространственная информация богата структурами высокого уровня, и не смотря на то, что каждая из классических моделей управления базами данных, которые стали появляться в 1960-х годах, смогла предложить в этой предметной области что-то свое, ни реляционная, ни объектно-ориентированная модель не согласуется с такой информацией в полной мере. Реляционная модель эффективно работает с топологическими отношениями, однако страдает от отсутствия средств представления сложных иерархических отношений, протяженных в пространстве. Объектно-ориентированные модели справляются как с топологическими, так и с иерархическими отношениями, но при этом испытывают трудности, имея дело с явлениями и вещами, которые характеризуются пространственной протяженностью.

В реляционной базе данных все объекты, сущности и понятия, которые отличаются друг от друга, представлены в виде отношений, или таблиц. Отношение определяется именем и перечнем различных атрибутов, которые его характеризуют, но при работе с пространственными данными необходим тип ломаных линий, чтобы привязать те или иные данные к географической точке или объекту. В реляционной БД заведомо этого типа данных нет, и есть необходимость создания множества таблиц и отношений для хранения пространственной информации, что создает сложности в работе с информацией.

Для обработки пространственной информации нужны более сложные конструкции, призванные уменьшить глубокое смысловое расхождение между точкой зрения пользователя и реализацией базы данных. Такие средства предложены в рамках парадигмы объектно-ориентированных программных средств, которые основаны на принципах пользовательских типов данных, а также наследовании и полиморфизме.

Дискуссии вокруг противопоставления реляционной и объектно-ориентированной модели ведутся в среде специалистов по базам данных параллельно со спорами относительно векторных и растровых элементов ГИС.

Географические информационные системы являются основной технологией, привлекающей внимание к системам управления пространственными базами данных (СУПБД, англ. SDBMS, Spatial Database Management Systems). ГИС предоставляют удобный механизм анализа и визуализации географических данных. ГИС обеспечивает богатый набор функций анализа, дающий пользователям возможность выполнять разнообразные преобразования географических данных. Великое множество технологий, встроенных в ГИС - это именно та причина, которая вызвала феноменальный рост этих систем и послужила причиной появления междисциплинарных приложений.

ГИС поддерживает большой набор операций над несколькими объектами и слоями, по сравнению с СУПБД, которая поддерживает более простые операции над совокупностью объектов и множеством слоев. СУПБД унаследовала от традиционных СУБД способность поддерживать механизмы управления параллелизмом, чтобы позволить множеству пользователей одновременно получать доступ к совместно используемым пространственным данным, сохраняя целостность этих данных.

ГИС может быть построена как интерфейсная часть СУПБД. Прежде чем ГИС сможет выполнить анализ пространственных данных, она обращается к соответствующим данным, находящимся в СУПБД. По этой причине эффективная СУПБД может существенно повысить эффективность и производительность ГИС. [14]

Геоинформационные системы начали свое развитие уже как больше чем пол века назад, вместе с развитием вычислительной техники. Сегодня прогресс прикладных программных технологий в сочетании с развитием телекоммуникаций, электронной техники, компьютерных сетей, средств обработки информации, методов управления базами данных и прикладного программного обеспечения способствовали тому, что за последние четверть века процессы дистанционного зондирования и мониторинга Земли обрели реальную эффективность в качестве профессиональных ГИС.

Если первые геоинформационные системы создавались поштучно, были дорогостоящими, сложными в обращении, имели весьма ограниченные возможности и не предназначались для широкого применения, то представители нынешнего поколения этих систем разрабатываются и выпускаются в промышленных масштабах множеством коммерческих компаний. Они характеризуются достаточной экономичностью и простотой эксплуатации, способны качественно выполнять самые разнообразные функции, пользуются активным спросом у корпоративных и индивидуальных потребителей и составляют в

совокупности значительный сегмент мирового рынка продукции информационных технологий. ГИС становятся чрезвычайно действенными и удобными в работе технологическими инструментами, позволяющими решать всё больше практических задач. Будь то спасение тонущего человека, или решение о выборе места строительства нового дома, или расчёта возможных убытков от техногенных катастроф. [16]

По функциональным возможностям ГИС нельзя сравнить ни с экспертной информационной системой, ни с инженерной информационной системой. (Рисунок 1.3.2) [17]



Рисунок 1.3.2 Функциональные возможности ГИС

В традиционном понимании географическая информационная система – это информационная система, обеспечивающая сбор, хранение, обработку, доступ, отображение и распространение пространственно-координированных данных. ГИС содержит данные о пространственных объектах в форме их цифровых представлений (векторных, растровых, квадратомиических и иных). Сущность ГИС определяется совокупностью программных и информационных средств, обеспечивающих не только выше указанные сбор, хранение, обработку, доступ, отображение и распространение, но и математико-картографическое моделирование и образное интегрированное представление пространственных и соотнесенных

с ними атрибутивных данных для решения проблем территориального планирования и управления. Из этого можно сделать вывод, что ГИС развиваются на стыке многих научных дисциплин и используются в разных областях управленческой деятельности.

Для полного представления всего разнообразия геоинформационных систем необходимо провести классификацию по степени пространственного доступа к данным и функциональности (Таблица 1.3.1). Различают: универсальные, специализированные, распределенные, мобильные.

Таблица 1.3.1 Классификация ГИС

	Универсальные/ специализированные/ мобильные/ распределенные	Поддерживаемая ОС	Разработчик	Формат БД
ARC/INFO	Универсальные	UNIX, Mac OS, Windows	ESRI	Oracle, SQL, Server, DB2
MapInfo	Универсальные	Windows, Linux	MapInfo	Access, Microsoft Excel, Oracle, Microsoft SQL Server, PostGIS, SQLite, любые БД поддерживающие ODBC
AutoCAD Map	Универсальные	Windows	Autodesk	Oracle, Microsoft SQL Server,
«Нева»	Универсальные	Windows	Институт им.В.А.Трапезникова	Access
dKart Hydrographer	Специализированные	Windows	Российская компания Моринтех (С-Map)	MS SQL
«Карта 2011»	Специализированные	Windows XP, 2003, Solaris, Linux, OC-PB 2000, MC BC, QNX, Pocket PC 2003	ЗАО «КБ «ПАНОРАМА»	Oracle, MS SQL Server, Firebird, MS Access черезODBC, ADO, BDE
GeoMedia Web Map	Распределенные	Любая, работа через браузер	Intergraph	Access, SQL Server, Oracle через ODBC

	Универсальные/ специализированные/ мобильные/ распределенные	Поддерживаемая ОС	Разработчик	Формат БД
ModelServer Discovery	Распределенные	Любая, работа через браузер	<u>Bentley Systems, Incorporated</u>	Связь через ODBC с Oracle
MapXtreme	Распределенные	Любая, работа через браузер	MapInfo	Связь с контейнерами данных через ODBC, DAO, OLE
ArcPad	Мобильные	Windows Mobile, Android, Apple	ESRI	Oracle, SQL, Server, DB2

Одними из самых распространенных ГИС являются **универсальные**, или многофункциональные. Они на сегодняшний день становятся неотъемлемой частью автоматизированного рабочего места любого специалиста: текстовый и графический редакторы, базы данных, электронные таблицы и прочее. К данному классу систем можно отнести такие программные продукты как ARC/INFO, MapInfo, AutoCAD Map.

Структура ГИС ARC/INFO включает несколько интегрированных модулей, предназначенных для работы с векторной графической информацией (точки, ломаные линии, замкнутые контуры), так и с атрибутивной (текстовой) информацией, логически связанной с географической.

Способ организации данных в ГИС MapInfo несколько отличается. В ней не поддерживаются основные топологические условия ARC/INFO, а картографическая информация в пределах слоя организуется в виде отдельных графических примитивов - объектов.

AutoCAD Map позволяет обрабатывать CAD-данные, более эффективно и наглядно решать практические задачи, классифицировать объекты, представляет расширенные возможности управления файлами реализации и их анализа. Инструменты модуля COGO, который используется для ввода данных, необходимы при создании высокоточных карт и особенно эффективны при вводе геоданных. Все объекты карты, содержащие атрибутивные данные в формате DWG, могут храниться в базе данных Oracle Spatial. Конверторы файлов векторных форматов позволяют обращаться ко всем источникам данных, адресуя запрос с использованием ключей для интеграции соответствующих векторных данных.

Так же к универсальным можно отнести ГИС «Нева» отечественной разработки. Эта ГИС предназначена для создания и обработки векторных, растровых, матричных карт и подготовки их к оперативному изданию цифровыми методами.

Кроме многофункциональных ГИС, существуют узкоспециальные геоинформационные системы, применяемые в специфических предметных областях и требующие специального оборудования и методов обработки данных. Такие системы называют **специализированными**.

Одной из таких предметных областей, где характерно использование специальных ГИС является сфера морской деятельности. Одним из представителей данного класса является ГИС dKart Hydrographer, которая предназначена для гидрографической съемки с целью составления морских карт, как векторных электронных, так и традиционных бумажных.

Так же использование ГИС широко распространено в сфере геодезических работ и строительных изысканий. Одним из программных продуктов, созданных для этой сферы деятельности является профессиональная ГИС «Карта 2011». Эта система комплектуется блоком геодезических расчетов, предназначенным для обработки данных, нанесения результатов вычисления на электронную карту и формирования отчетных документов по метрическим и атрибутивным данным.

По мере роста объема задач, реализуемых ГИС, и совершенствования базовых технологий, стало возможным распределять геоданные в сетях с одновременным созданием интерактивных сред взаимодействия клиента приложений с геоинформационным сервером и создавать **распределенные** ГИС. Технология взаимодействия с базой данных является наиболее критичной с точки зрения быстродействия геоинформационного WEB-сервера. Обычно такие сервера работают со структурами информации родственных ГИС-систем и только некоторые поддерживают форматы других ГИС.

Наибольшую известность среди разработчиков экспертных систем из распределенных ГИС получил комплекс продуктов под названием MGE (Modular GIS Environment). Все данные ГИС MGE располагаются в MGE-проекте. Для публикации геоинформационных данных, содержащихся в проекте, разработан продукт под названием GeoMedia Web Map. GeoMedia Web Map позволяет публиковать данные проекта, просто файлы в формате DGM, ARC/INFO, ARC/View. Для просмотра этих данных на клиентском месте требуется любой браузер и необходимо поставить дополнительный plug-in, который позволяет просматривать active CGM (Рисунок 1.3.3).

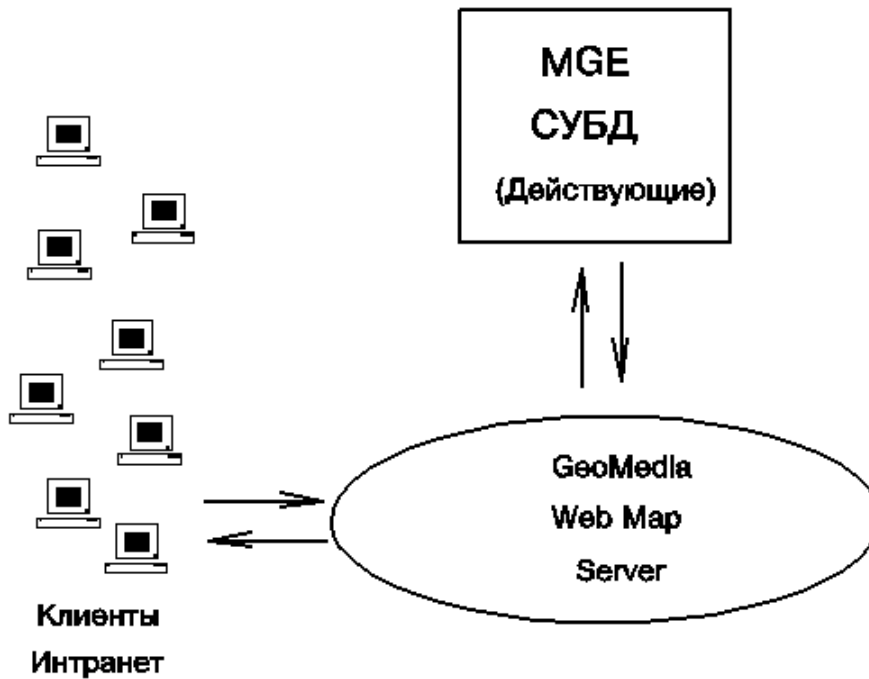


Рисунок 1.3.3 Типовая конфигурация работы Web-сервера на основе GeoMedia Web Map

Также к распределенным ГИС следует отнести программный продукт ModelServer Discovery, который предназначен для публикации геоинформационных данных в корпоративных распределенных средах. Особенностью данного продукта является то, что он позволяет публиковать не только данные родственной ГИС в среде GeoGraphics, но и данные любого другого проекта, базирующегося на технологии DGN-файлов (Рисунок 1.3.4).

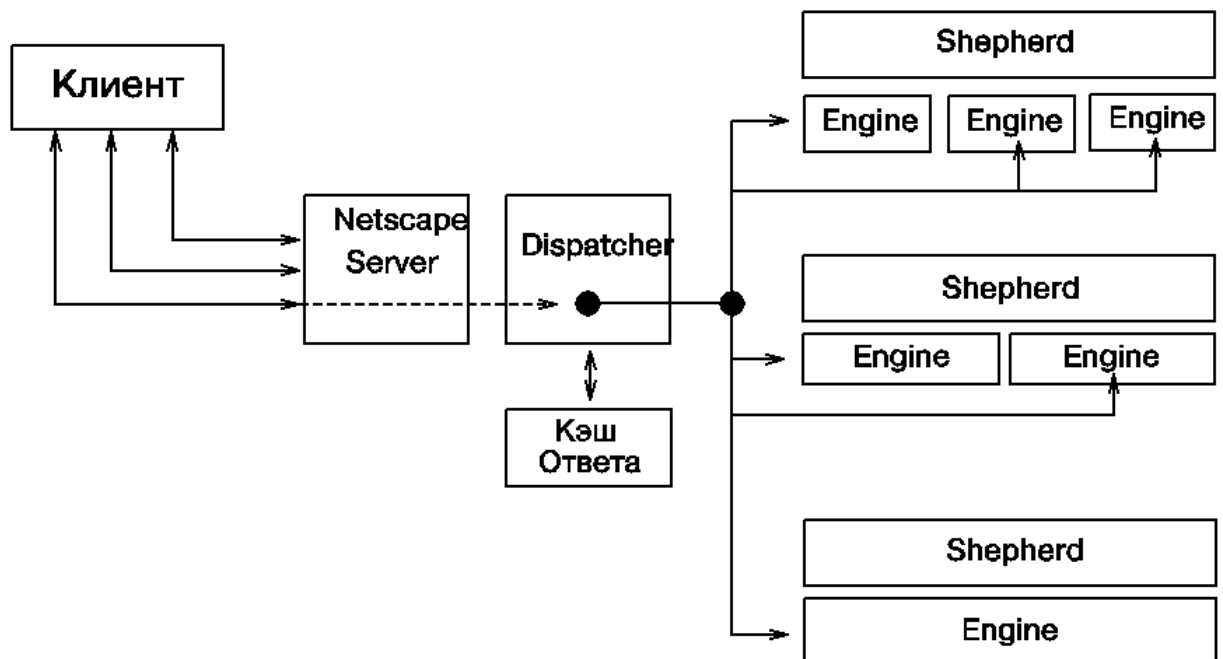


Рисунок 1.3.4 Принцип работы ModelServer Discovery

ModelServer преобразует исходные данные геоинформационного проекта в формат, запрошенный клиентом. ModelServer состоит из 3-х модулей:

- Dispatcher - представляет собой plug-in в Netscape Enterprise Server. Он отвечает за взаимодействие сервера Netscape с внутренними процессами ModelServer-a. Получив запрос, данный модуль передает его дальше в модуль Shepherd.
- Shepherd - организует управление процессами Engine. Следит за запуском, перезапуском данных процессов.
- Publishing Engine - производит преобразование исходных геоинформационных данных в формат, затребованный пользователем. Так же он отвечает за реализацию ряда функций пространственного запроса:
 - Подсоединить/отсоединить карты
 - Включение/выключение карт
 - Географическая локализация
 - Пересечение объектов
 - Пространственный поиск

После того, как модуль Dispatcher получает от модуля Engine информацию в требуемом формате, он посылает ее через сервер Netscape пользователю. Так же он сохраняет эту информацию в кэш. В случае повторного запроса на данный объект, информация берется из кэша.

Фирма MapInfo известна в мире как производитель настольной ГИС MapInfo, очень популярной в широких кругах. Производимый ими продукт MapXtreme предназначен для публикации данных ГИС формата MapInfo в WWW (такими данными могут являться таблицы MapInfo, растровые изображения, карты улиц и пр.). Применяемая в MapXtreme технология визуализации динамических геоинформационных данных базируется на использовании стандартных тэгов языка HTML (Рисунок 1.3.5). Это позволяет в отличие от рассмотренных выше систем использовать для просмотра данных действительно любой клиентский браузер на платформе любой операционной системы. Также отпадает необходимость установки на клиентской машине каких – либо модулей расширения (plug-in).

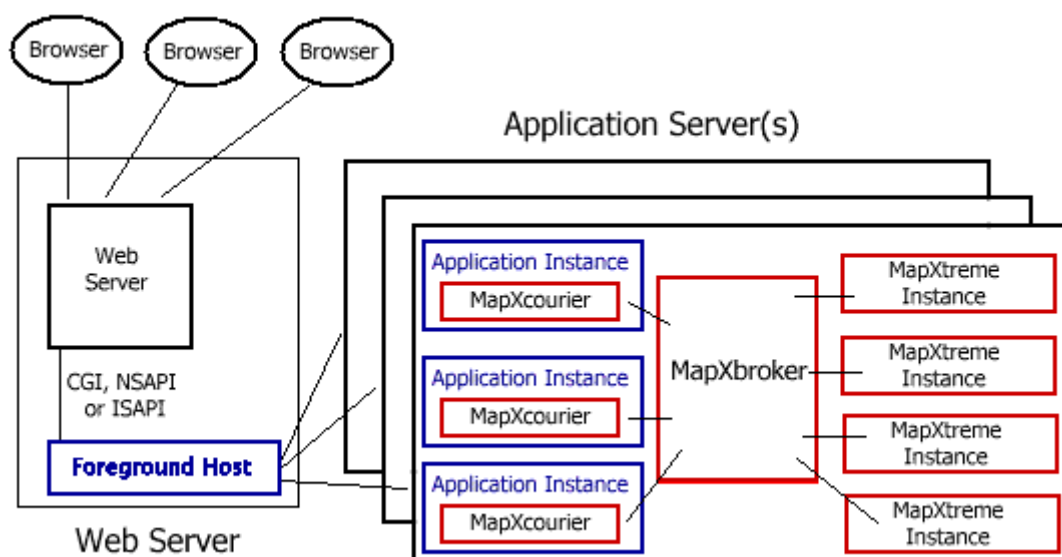


Рисунок 1.3.5 Принципы работы MapXbroker

1. Web server может быть любым WWW-сервером под управлением MS Windows или Unix.
2. Application Server(s) предназначен для обработки всех запросов пользователя к графическим или атрибутивным данным и обеспечения безопасности. Он может быть установлен на том же компьютере, что и Web server или на одном или нескольких выделенных.
3. Foreground Host перераспределяет запросы от Web server к Application Servers.
4. Application Instance создается для каждого запроса в виде отдельного процесса в многозадачных операционных системах. Эти процессы используют интерфейс MapXcourier для доступа к MapXtreme.
5. MapXbroker получает запросы от MapXcourier и передает их следующему доступному MapXtreme Instance.
6. MapXtreme Instance выполняет всю работу, связанную с обработкой конкретного запроса и созданием выходных данных для отсылки их обратно клиентскому браузеру. MapXtreme Instance является предстартовой службой и поэтому при поступлении каждого нового запроса не тратит время на инициализацию и на подключение к удаленным базам данных. [18]

Дальнейшее развитие современных технологий позволяет прогнозировать бурное развитие распределенных ГИС. Особенное влияние окажет расширение коммуникационных возможностей и увеличение пропускной способности систем связи. Наиболее важным проектом в сфере распределенных ГИС является Интернет-решение сервера картографической информации Google maps. Появление в 2005 году новых веб-сервисов Google Maps и Google Earth не только открыло доступ к качественным современным геоданным, но и позволило всем

желающим создать свои собственные геоинформационные подсистемы. Первые ГИС на платформе Google Earth уже используются метеорологами, геологами и археологами и многими другими.

Одним из перспективных направлений применения распределенных геоинформационных систем является создание мобильных ГИС. В рамках создания каждой из реализаций такой ГИС выполняются работы по сбору и актуализации пространственных данных. Для достижения заданной точности определения положения объектов инфраструктуры, обеспечения актуальности данных, создавать и модифицировать пространственные данные необходимо непосредственно на объектах.

В настоящее время GPS приемники позволяют проводить полевые работы в любых погодных условиях, в любое время дня и ночи. При этом точность определения местоположения составляет от 0,5 до 50 м. Мобильные ГИС с приемником GPS позволяют непосредственно в полевых условиях: обновлять существующие ГИС данные, проводить сбор данных для ГИС, осуществлять навигацию по карте.

Система ArcPad, одна из представительниц мобильных ГИС, является частью платформы ГИС предприятия и интегрируется непосредственно с ArcGIS для Desktop, ArcGIS для Server, и ArcGIS Online.

Использование мобильных ГИС существенно повышает качества наблюдения за пространственными процессами, позволяя уточнять специальную информацию (картографическую или объектовую), координировать усилия мобильных служб и своевременно принимать меры по устранению угроз. Слияние технологий ГИС, GPS, GPRS позволяет определять местоположение, прокладывать оптимальный маршрут движения, производить контроль за движением объектов, их поиск и многое другое. Типичными представителями мобильных ГИС являются всем известные навигаторы. [16]

Все рассмотренные выше распределенные и мобильные ГИС работают непосредственно с централизованной базой данных или сервером с единой базой данных.

В последнее время наблюдается тенденция объединения институтов, крупных корпораций и производств для совместной деятельности и соответственно возникает необходимость объединения ресурсов, в том числе и информационных. Построение геоинформационной системы для совокупности нескольких организаций может осуществляться по двум направлениям.

Первый заключается в создании абсолютно новой ГИС, где все информационные потоки и организация взаимодействия данных будет построена по созданной оптимальной схеме независимо от уже существующих. Проектирование будет заключаться в создании

централизованной базы данных на сервере организации и установки клиентских приложений на рабочих местах сотрудников.

Второй подход предусматривает использование накопленных данных и отрегулированных систем связи между организациями или внутри организации для построения распределенной ГИС с применением распределенной СУБД. Для проектирования такой системы необходимо организовать доступ к разнообразным базам данных, имеющим совершенно разные форматы и структуру представления данных.

Наиболее простым и часто используемым является первый подход, так как создание новой информационной системы, не привязанной к существующим в организации связям, всегда предусматривает нахождение стандартных технологий решения проблем. Но практика показывает, что накопленные данные всегда важнее, чем создание благоприятных условий разработчикам.

1.4 Технологии построения информационных систем на основе гетерогенных баз данных

Бурное внедрение информационных технологий в деятельность организаций привело к развитию распределенных информационных систем (РИС), реализующих различные информационные среды на основе разнообразных баз данных, управляемые как современными, так и «морально» устаревшими системами управления базами данных (СУБД). Это привело к появлению неоднородного информационного пространства, реализующего гетерогенные базы данных. К управлению такими распределенными системами предъявляются повышенные требования с точки зрения обеспечения целостности и непротиворечивости данных, степени однородности информационного пространства. Проблема состоит в технологической сложности перехода от централизованного однородного управления данными к распределенной обработке данных в гетерогенной информационной среде, включающей автономные системы работы в локальной сети гетерогенных баз данных, не представляющие возможность контроля за выполнением глобальных транзакций.

Существующие разработки в этой области не позволяют эффективно решить задачу одновременного управления несколькими автономными системами гетерогенных баз данных. Некоторые программные продукты предлагают использование шлюзов для доступа к другим системам ГБД с ограниченными возможностями, требующие полного изменения организационной структуры уже существующих информационных систем. Средствами, базирующимися на использовании шлюзов, свойственны некоторые серьезные ограничения.

Во-первых, шлюзы не дают возможности организовать систему управления транзакциями даже для отдельных пар систем. Во-вторых, использование шлюзов призвано лишь решить задачу трансляции запросов с языка одной СУБД на язык другой. При этом, в качестве средства управления несколькими системами ГБД, используются мониторы транзакций. Среди недостатков мониторов транзакций следует указать ограниченное количество поддерживаемых ими сетевых протоколов и СУБД. Те СУБД, которые используются в узлах системы, должны поддерживать протокол двухфазной фиксации и удовлетворять стандарту X/OpenXA, описывающему особенности управления транзакциями, работающими с несколькими гетерогенными СУБД. Отсюда следует, что разработка эффективных моделей, алгоритмов и программных средств ориентированных на обеспечение интегрированного доступа к гетерогенным базам данных является актуальной задачей.

Перед появлением систем баз данных вся информация в компьютерах хранилась в гетерогенных файлах, создаваемых различными автономными приложениями. Файловые системы были разработаны из-за потребности получения более эффективных доступов к данным. Но не все предприятия и организации начали освоения информационных технологий по логически правильному пути создания единого централизованного хранилища данных, а создавались базы данных разделенные по областям деятельности.

Развитие экономики и создание транснациональных корпораций также оказало большое влияние на развитие распределенных информационных систем и систем управления гетерогенными базами данных. Существовала необходимость объединения разрозненных данных, имеющих различия в наименовании полей, их типов и файловой структуре.

Для решения данной проблемы в начале семидесятых годов было введено новое понятие – база данных (БД).

Развитие сетей передачи данных и распределенных вычислений в конце восьмидесятых годов обеспечило технический базис для разработки распределенных баз данных (РБД). Распределённая база данных – это распределённые данные и связи между ними. [19] В качестве преимущества использования распределенных СУБД можно отнести возможность отражения структуры организации, разделяемость и локальную автономию, повышение доступности данных, повышение надежности и производительности, модульность системы. [19,20,21,22,23] Но широкое применение баз данных для хранения и обработки информации в производственной и непроизводственной сфере привело к огромному числу независимых источников информации, управляемые различными СУБД. Каждая база данных создавалась индивидуально и независимо друг от друга, каждая для своих целей, поэтому не удивительно, что они различаются по физической и логической структуре, имеют собственную схему, языки запросов. Это было обусловлено естественной эволюцией рынка аппаратных платформ и

средств хранения. Поэтому вполне объяснимо соседство современных компонентных многоуровневых систем, клиент серверных технологий и настольных СУБД.

Заложенные в основе традиционных параллельных и распределенных СУБД принципы однородности и централизованного управления не позволяют обрабатывать автономные данные, созданные в различных программных средах. Для работы в распределенных информационных системах с гетерогенными базами данных требуется новая распределенная архитектура, которая позволяла бы обеспечить управление различными базами данных без потери их автономности и коренной перестройки архитектуры существующих локальных информационных систем в целом. Поддерживала бы возможность концентрации информации с целью оперативного анализа, долгосрочного планирования и прогнозирования явлений, угрожающих благосостоянию организации, региона или страны. Такая архитектура должна обеспечить свойства распределенных баз данных, выделенных C.J.Date в 1987 году [31]:

- Локальная автономия (local autonomy);
- Независимость узлов (no reliance on central site);
- Непрерывные операции (continuous operation);
- Прозрачность расположения (location independence);
- Прозрачная фрагментация (fragmentation independence);
- Прозрачное тиражирование (replication independence);
- Обработка распределенных запросов (distributed query processing);
- Обработка распределенных транзакций (distributed transaction processing);
- Независимость от оборудования (hardware independence);
- Независимость от операционных систем (operationg system independence);
- Прозрачность сети (network independence);
- Независимость от баз данных (database independence).

В середине восьмидесятых годов для выполнения всех этих свойств было предложено ряд архитектур, характеризующиеся различной степенью автономности и гетерогенности локальных узлов [24,25,26]: федеративные системы, мультибазовые системы. На рисунке 1.4.1 представлены направления обеспечения управления базами данных относительно трёх измерений: распределенность, автономность, гетерогенность.

Автономность является наиболее важной и необходимой характеристикой при построении современных распределенных информационных систем. Автономность показывает степень независимость отдельных СУБД от системы в целом. Это функция множества факторов, а именно возможности обмениваться информацией, возможности полного контроля над всеми транзакциями выполняющимся в СУБД, возможности внесения изменения во внутреннюю структуру СУБД.

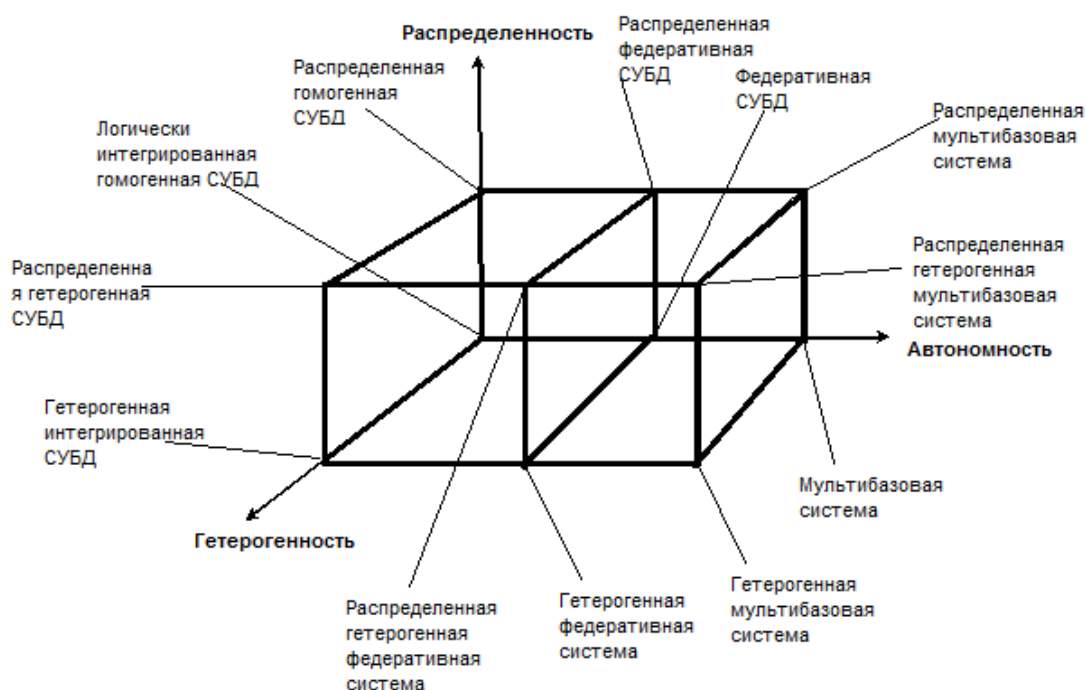


Рисунок 1.4.1. Направления обеспечения управления базами данных

Гетерогенность СУБД означает различие средств и методов разработки, поддерживание различных программных и аппаратных платформ, коммуникационных протоколов, применяемых механизмов конкурентного управления транзакциями. Главной же целью построения распределенной информационной системы является скрытие от пользователя тот факт, что система состоит из гетерогенных СУБД, и сохранить их автономность.

Под гетерогенной базой данных понимается совокупность разноформатных баз данных, с различными таблицами, полями, атрибутами, представлениями данных и их структурой.

Ещё одна проблема при построении РИС является физическая распределенность данных. С ней в настоящее время справляются по средствам технологии CORBA.

Сейчас всё больше разработчиков интересуются распределенными информационными системами на основе гетерогенных баз данных, их ещё называют распределенными гетерогенными мультибазовыми системами. Предпринимаются попытки интеграции таких распределенных систем ГБД, в которых весь контроль над отдельными локальными системами полностью осуществляется их операторами. А так как узлы полностью автономны, то не возникает необходимости внесения каких либо изменений в локальные СУБД. Это достигается путем создания поверх существующих локальных систем дополнительного уровня программного обеспечения (Рисунок 1.4.2).

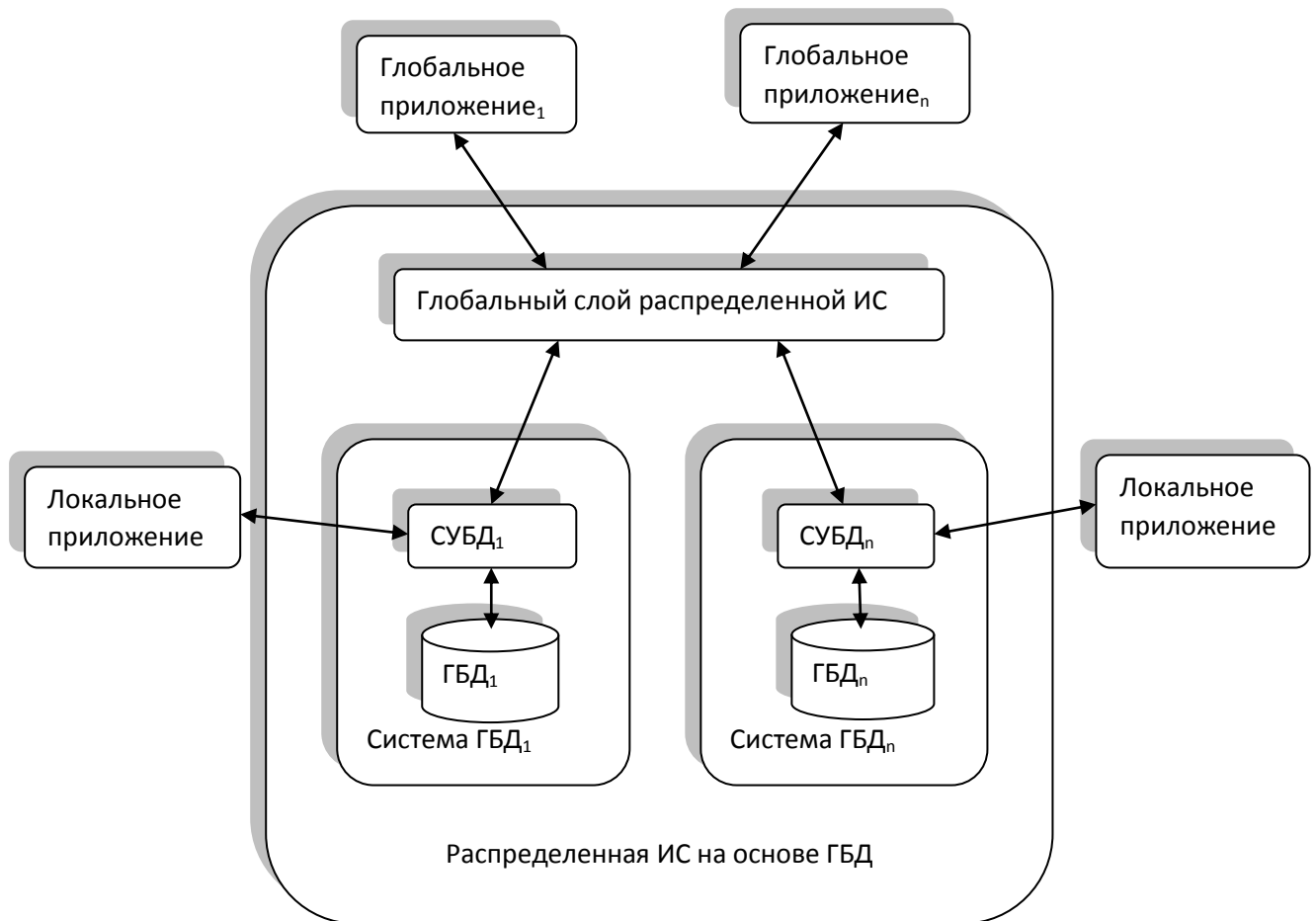


Рисунок 1.4.2 Модель распределенной информационной системы на основе гетерогенных баз данных

Преимущества данного подхода заключаются в следующем [30]:

1. возможность создания интегрированной системы на базе уже существующих информационных систем;
2. сохранение инвестиций в существующее оборудование, системное и прикладное ПО;
3. сохранение локальной автономии с возможностью доступа к локальным данным глобальных пользователей распределенной ИС.

Наравне с построением РИС на основе гетерогенных БД, используется другой подход, основанный на применении шлюзов. Он заключается в применении шлюзов для отдельных частей гетерогенной РИС, предназначенных для преобразования языка и модели данных каждого из используемых типов СУБД в язык и модель данных реляционной системы. Но этот подход осуществим только при реляционной структуре построения баз данных.

Использование шлюзов не позволяет справиться с проблемами вызванными неоднородностью структуры и представлением данных в различных схемах [27]. Шлюзы можно назвать средствами, облегчающими миграцию, но никак не средствами интероперабельности в распределенных системах.

Следовательно, можно сделать вывод, что развитие подходов к созданию распределенных информационных систем на основе гетерогенных баз данных идет по пути организации децентрализованной структуры. С ростом количества гетерогенных источников информации, задачи построения и исследования подобных распределенных систем приобретает все большее значение при одновременном усложнении самой задачи.

При построении РИС на основе гетерогенных БД необходимо решить главные три задачи:

1. выбрать механизм интеграции схем различных гетерогенных база данных;
2. выбрать модели взаимодействия компонентов;
3. выбрать протокол управления глобальными транзакциями в распределенной системе.

Для выбора механизм интеграции схем необходимо ознакомиться с существующими.

Традиционная трёхуровневая архитектура ANSI SPARC (комитета планирования стандартов Американского национального института стандартов), используемая для описания централизованных систем баз данных, представлена на рисунке 1.4.3.

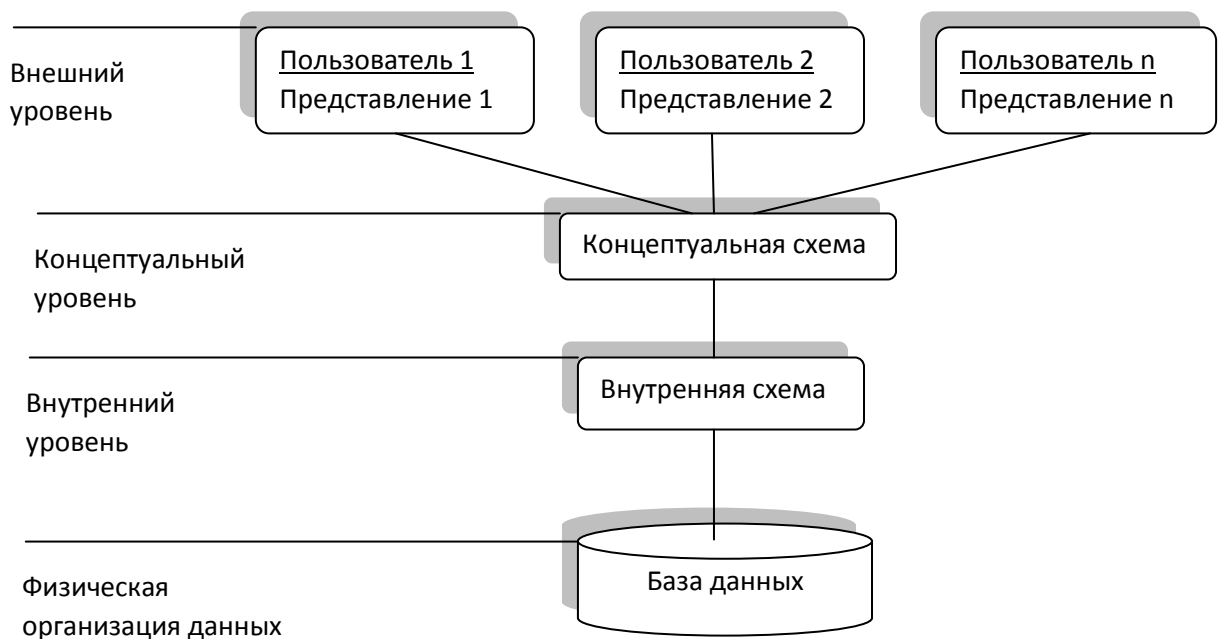


Рисунок 1.4.3 Трёхуровневая архитектура ANSI SPARC

Цель разработки данной архитектуры заключается в отделении пользовательского представления БД от её физического представления. Она может быть расширена на случай использования РИС на основе гетерогенных баз данных и будет выглядеть как представлено на Рисунке 1.4.4.

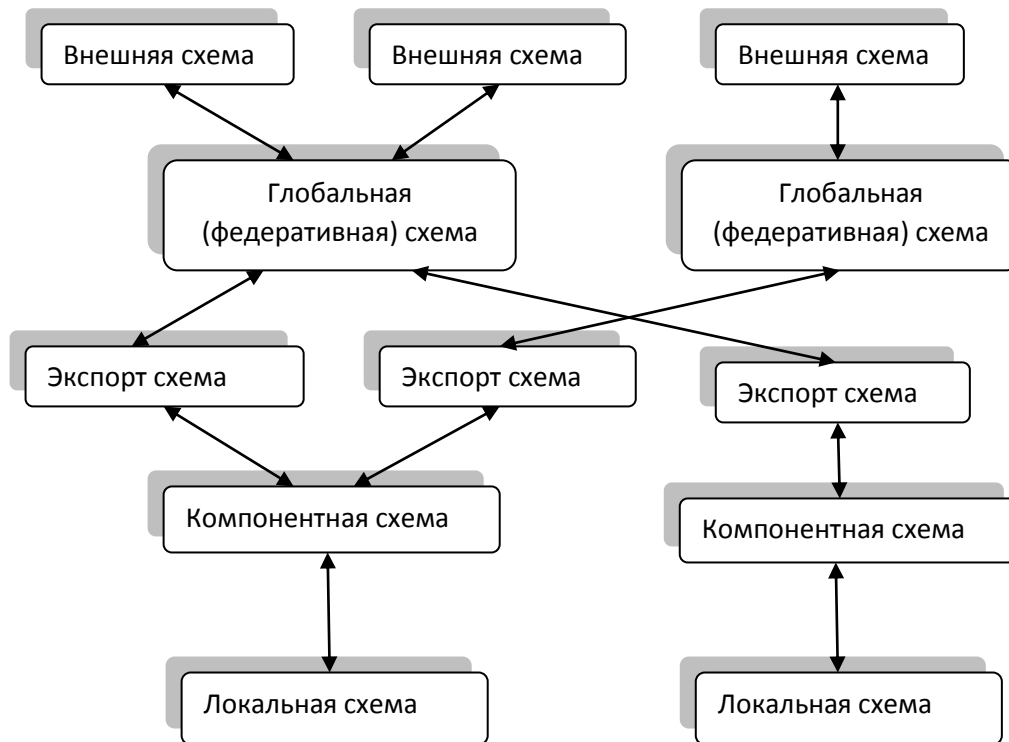


Рисунок 1.4.4 Пятиуровневая архитектура РИС на основе гетерогенных баз данных

Локальная схема выражается в модели данных соответствующей базы данных, поэтому локальные схемы различных гетерогенных БД могут быть выражены по средствам различных моделей данных. Для обеспечения доступа к системе локальные схемы транслируются в компонентные схемы на языке общей модели данных.

В качестве общей модели данных могут быть использованы реляционная, объектно-реляционная или полуструктурированная модель. Но при выборе общей модели необходимо руководствоваться критерием совместимости общей модели данных с моделями данных интегрируемых компонентов.

Каждая гетерогенная база данных присутствует в системе посредством экспорта части своей компонентной схемы или экспорт схемы. Глобальная схема создаётся путем интегрирования множества экспорт схем и используется, когда имеется только одна такая схема. Федеративных схем может быть несколько. А внешняя схема создаётся для контролирования доступа пользователей к определённым объектам базы данных. [19]

Выделяют два основных типа распределённых систем на основе ГБД в зависимости от различных уровней интеграции экспорт схем [28]:

1. без поддержания глобальной схемы;
2. с поддержанием глобальной схемы.

Подход без поддержания глобальной схемы предполагает отсутствия интеграции экспорт схем, его поддерживают такие программные продукты как: Oracle v5, SybaseCalifornia Intelegent Database Assistant (CALIDA), DB2 DataJoiner.

Важным компонентом таких систем является универсальный язык запросов интегрируемых гетерогенных БД в распределенную информационную систему. Построение РИС с использованием унифицированного языка запроса возможно только тогда, когда сами источники предлагают язык запросов [29]. Данный подход обеспечивает автономию локальных узлов улучшает гибкость РИС в целом. Особенностью архитектуры такой распределенной информационной системы является комбинация экспорт схем на различных её уровнях (Рисунок 1.4.5).

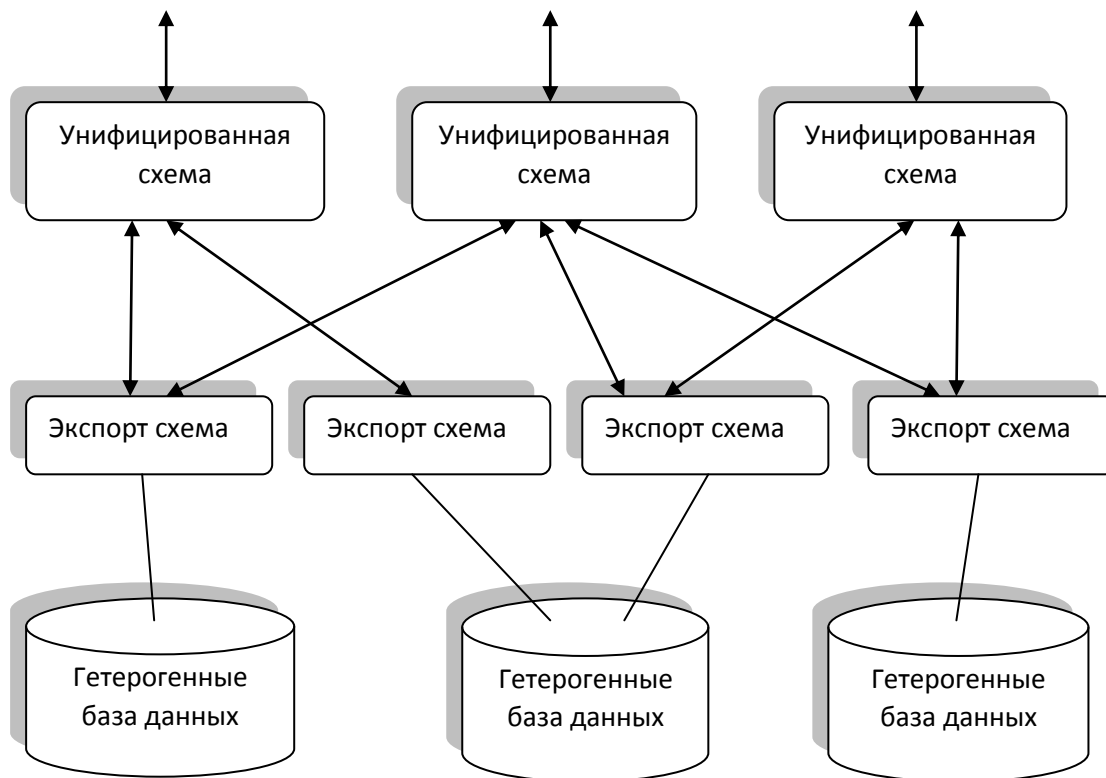


Рисунок 1.4.5 Архитектура распределенной информационной системы без глобальной схемы

Подход на базе глобальной схемы предполагает объединение экспорт схем, что позволяет интегрировать автономные гетерогенные БД в логически единую базу данных.

Программное обеспечение предварительно транслирует глобальные запросы и превращает их в запросы и операторы модификации данных соответствующих локальных СУБД. Тем самым полученные результаты локальных запросов сливаются в единый глобальный результат, который и представляется пользователю. Также, в данной модели, происходит контроль за выполнением фиксации или отката отдельных операций глобальных транзакций в глобальных СУБД, тем самым обеспечивая целостность данных в каждой локальной базе данных. Главная задача, решаемая при использовании глобальных схем, заключается в управлении логической гетерогенностью в схемах источников данных. [30] Среди программных продуктов поддерживаемых эту технологию являются: MERMAID [29], DATAPLEX, MULTIBASE, UniSQL/M.

Управление распределенной информационной системой намного упрощается при использовании реляционных гетерогенных баз данных, так как в этом случае отпадает необходимость трансляции локальных схем в компонентные схемы (Рисунок 1.4.6). [30] Но на практике чаще встречаются разнородные модели в составе гетерогенных баз данных.

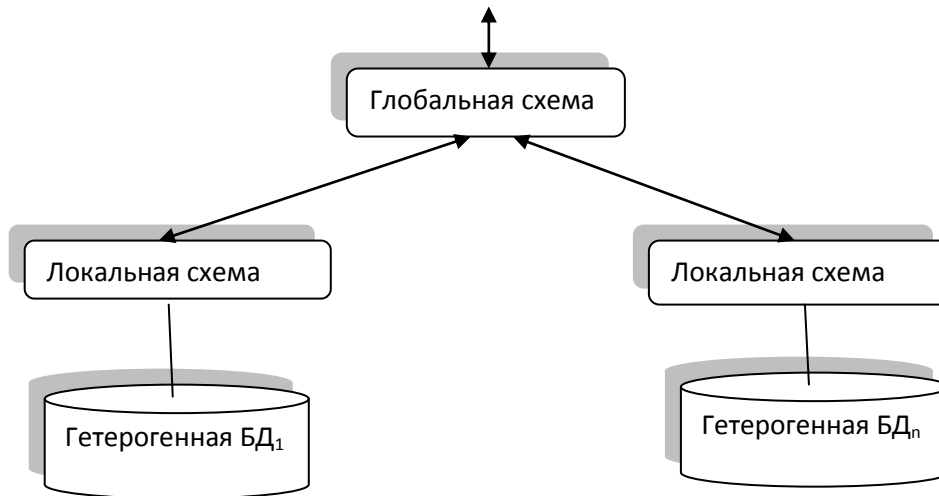


Рисунок 1.4.6 Архитектура распределенной информационной системы на основе глобальной схемы

Одним из недостатков глобальной схемы является невозможность гарантировать полную автономность локальных баз данных. В этом подходе обязательно определение модели данных и языков запросов, используемых в глобальной схеме. Каждый запрос к гетерогенной базе данных выполняется в глобальной схеме по средствам языка запросов, чаще всего используют SQL. Последующее разбиение глобального запроса на подзапросы к локальным базам данных является трудоемкой задачей. Существует и другие проблемы работы с глобальной схемой. Одной из них является добавление новых гетерогенных баз данных в распределенную ИС и модификация существующих локальных схем. В любом случае применение данного подхода может вести к изменению всей структуре распределенной информационной системы. [28]

Вывод по главе

В соответствии со «Стратегией развития Арктической зоны Российской Федерации и обеспечения национальной безопасности на период до 2020 года», утвержденной президентом Российской Федерации В.В. Путиным, одним из факторов оказывающим влияние на социально-экономическое развитие Арктической зоны являются экстремальные природно-климатические условия, включая низкие температуры воздуха, сильные ветры и наличие ледяного покрова на акватории арктических морей. А также недостаточное развитие навигационно-

гидрографического и гидрометеорологического обеспечения мореплавания, что сказывается на уровне освоения территорий и экономических рисках освоения территорий.

Для создания эффективной геоинформационной системы управления обширными территориями необходимо придерживаться требований вытекающих из структуры пространства решений. Наиболее значимыми в формировании требований к созданию актуальной системы управления развитием территории на основе ГИС является пространство состояний объектов управления и пространство решений, характеризующие основную деятельность при работе с информацией.

Но ни одна из рассмотренных технологий не может удовлетворить требованиям создания распределённой вычислительной системы обработки геоданных с использованием разноформатных баз данных, так как существуют определённые особенности в представлении и обработки географических данных и связанных с ними представлений.

После проведенного анализа телекоммуникационных систем распространения пространственно-временной геоинформации было выделено два подхода к созданию единой распределенной геоинформационной системы и объединению тем самым различных баз хранения информации.

Однако способы организации подобной системы ещё находятся на стадии зарождения и имеют большое количество проблем и незавершенных вопросов.

Следующим шагом в освоении технологий работы с разноформатными распределенными базами данных и её применением в создании геоинформационных систем будет моделирование новых архитектур подобных систем на основе уже существующих.

2. Разработка модели геоинформационной системы на основе гетерогенных баз данных

Проектирование компьютерных информационных систем невозможно представить без моделирования. Оно стало составной частью общих подходов, применяемых в современных информационных технологиях. Особенно стоит подчеркнуть, что именно моделирование стало тем связующим звеном между формальным и неформальным мышлением. А с помощью компьютерной техники построение моделей систем вывело человечество на новый уровень информационного прогресса. Современные средства интерфейса и программирования дают возможность исследователям задавать компьютеру вопросы, просчитывать альтернативы решений и проверять гипотезы, что позволяет исключить из действительности опасные явления и действия, которые только могли бы произойти.

2.1 Модели информационно-управляющих систем

2.1.1 Общие понятия моделирования

В 21 веке нельзя назвать область человеческой деятельности, в которой в какой-либо мере не использовались методы моделирования. Особое место занимает проектирование различных систем и управление ими, где основным является принятие управленческих решений на основе полученной или представленной определенным образом информации.

Моделирование можно назвать одним из основных методов исследования сложных систем, который используется для принятия управленческих решений в различных сферах науки и техники.

Идея моделирования состоит в замене реального объекта некоторой моделью, прообразом этого объекта. При этом у исследователя появляется возможность проводить эксперименты, которые были бы невозможны над реальным масштабным объектом. Так работа не с самим объектом или явлением, процессом, а его моделью происходит достаточно быстро и без существенных материальных затрат на его исследование в различных ситуациях. Однако вычислительные эксперименты с моделью системы позволяют изучить её поведение в достаточном объеме, что не позволяют теоретические подходы.

Методология моделирования предполагает тщательную разработку модели, начиная с самой простой модели, заканчивая моделью сложной системы, таким образом, продвигаясь к более совершенному описанию реальности.

В соответствии с физическим содержанием модели их можно разделить на натурную модель и абстрактную. Натурная модель – та модель, которая выполнена из физического материала, а абстрактная – та, которую «нельзя потрогать».

Моделирование – это представление системы моделью для получения информации об этой системе путем проведения экспериментов с этой моделью [37,38].

Моделированием также можно назвать замещение одной системы другой для получения информации о важных свойствах системы-оригинала.

Наиболее актуальной проблемой в современной науке и технике считается разработка и применение методов проектирования и исследования сложных технологических систем, в состав которых также входят информационно-управляющие, информационно-вычислительные системы различного уровня. При создании подобных систем возникает множество вопросов, ответы на которые требуют проведение оценки количественных и качественных закономерностей процессов, проходящих в сложных системах, а также структурного и параметрического синтеза.

Все перечисленные выше системы относятся к классу сложных систем, проектирование, эксплуатация и эволюция которых невозможна без создания моделей.

Под сложными системами будем понимать системы, которые обладают большим числом элементов. Их свойства не могут быть предсказаны зная лишь свойства отдельных частей системы и способы их соединения.

Однако создание подобных систем имеют следующие особенности [39]:

1. Сложность структуры и стохастичность связей между элементами;
2. Неоднозначность алгоритмов поведения при различных условиях;
3. Большое количество параметров и переменных;
4. Неполнота и неопределенность исходной информации;
5. Разнообразие и вероятностный характер воздействий внешней среды и др.

Модели разрабатываются для выполнения каких-либо задач или решения проблем, которые стоят перед исследователем. Требования, предъявляемые к модели, зависят от характера задач, которые она решает. Во многих таких задачах модель служит, прежде всего, источником информации при выполнении вычислений для получения конечного результата. Созданные при её формировании математические модели являются лишь хранителем или носителем информации о связях между переменными, характером их изменения и свойствах системы.

Задачи, при решении которых наиболее часто используют модели, следующие [39]:

1. Проектирование различного рода сложных систем;
2. Оптимизация и оптимальное управление сложными системами;
3. Исследование свойств систем и процессов;
4. Прогнозирование поведения систем и процессов;
5. Реинжиниринг организации;
6. Планирование деятельности и ресурсов;
7. Принятие различного рода управленческие решения;
8. Проблемы изучения и использования искусственного интеллекта.

В данном перечислении наиболее подходящими по тематике данного исследования являются оптимизация и оптимальное управление различными сложными системами, прогнозирование поведения систем и процессов, планирование деятельности и ресурсов, проектирования различного рода сложных систем. Следовательно, можно однозначно утвердить необходимость построения различного рода моделей для построения распределенной системы управления Арктическими территориями РФ.

В первую очередь система будет информационной, построенной на основе компьютерной техники и новых информационных технологий, значит, есть необходимость в построении информационной модели и использовании методов компьютерного моделирования.

Информационной моделью является модель объекта, процесса или явления, включающая информацию в качестве основной составляющей моделируемого объекта, процесса или явления [40]. Компьютерное же моделирование можно обозначить как процесс моделирования реализуемое с помощью компьютерных средств. Его можно отнести к абстрактному моделированию, в отличие от информационной модели, которая может быть отнесена и к натурному и к абстрактному моделированию одновременно.

В прикладных областях науки можно выделить следующие виды абстрактных моделей:

1. Математические модели. Математическое моделирование отражает реальный объект, процесс, систему путём построения выражений, понятий, отношений с помощью системы математических символов и обозначений.
2. Графические модели. Графические модели решают задачу визуализации явлений и процессов. В научных исследованиях результат может быть сложным и абстрактным, и человек, в связи со своим психофизическим состоянием может не распознать научный результат, в чём ему помогают графические модели. Сегодня можно говорить о применении теории графов в создании программных продуктов помогающих в разработке информационных систем и называемых CASE-технологиями.

3. Вербальные или текстовые модели. Эти модели используют последовательности предложений на формализованных диалектах естественного языка для описания той или иной области реальности. К ним можно отнести любые описания, включая и набор текста.

4. Имитационные модели. Данные модели применяются для описания свойств реальной системы, получаемые из проведения экспериментов над этой моделью. Построение имитационной модели, наиболее часто встречаемое моделирование для получения необходимых результатов при построении эффективной и рентабельной системы. В последнее время имитационное моделирование производится с помощью компьютеров.

5. Модели информационных процессов описывают возникновение, передачу, преобразование и использование информации в самых разнообразных системах.

Информационные модели могут быть представлены как объекты и процессы в виде рисунков, схем, чертежей, таблиц.

Наиболее часто встречаемый вид информационной модели - таблица. С её помощью можно выразить статические и динамические информационные модели.

В табличной информационной модели объекты или их свойства представлены в виде списка. К ним можно отнести табличные редакторы и базы данных.

Состав и набор элементов информационной модели соответствует степени детализации описания состояний и функций выполняемых объектом управления. Таким образом любой элемент в информационной модели связан со свойством объекта управления.

В современной информационной системе кроме информационной модели будет присутствовать и модель данных.

Модель данных отображает сами данные и их связи. До недавнего времени выделяли три вида модели данных: иерархическая, сетевая и реляционная. В современном мире также выделяют такие модели как постреляционные, объектно-ориентированные и другие. Данные модели непосредственно связаны с базами данных и системами их управления, в которых строятся модели процессов накопления и применения информации и знаний. Для формирования модели данных используются [40]:

1. Структурная схема объекта;
2. Структурно-функциональная схема объекта;
3. Алгоритмы необходимые для функционирования системы;
4. Схема расположения технологических средств на объекте;
5. Схема связей и так далее.

Для создания модели данных используется анализ потоков данных, который можно провести с помощью метода DFD или IDEF0 диаграмм, а также методов интеллектуального анализа. Применение подобных методов позволяет на основе накопленной информации принимать необходимые решения и производить новые знания, которые способны повысить эффективность решений и деятельность людей или предприятий.

Тем самым сбор и анализ реальных данных полученных при функционировании системы или объекта моделирования дает количественные оценки для разработки вариантов программно-технического обеспечения автоматизированных систем.

Для проектирования любой информационной системы применяют системный подход, который позволяет рассматривать различные объекты с единой точки зрения, при этом показывая важнейшие черты их функционирования и учитывая наиболее существенные факторы. При использовании системного подходы обычно говорят о применении структурного подхода к проектированию программного обеспечения, так как любая система имеет структуру.

Сущность структурного подхода для проектирования программного обеспечения автоматизированной системы управления Арктическими территориями РФ заключается в её декомпозиции, разбиении на автоматизируемые функции и уровни, если она имеет многоуровневую архитектуру. При этом система делится на функциональные подсистемы, которые в свою очередь делятся на подфункции, а те делятся на задачи, и так далее. При этом сама система сохраняет целостное представление.

В методологии структурного подхода используют принципы:

1. Сложные задачи разбиваются на множество меньших подзадач, решения которых будет легким для понимания;
2. Создание иерархической древовидной структуры из составных частей задачи с добавлением новых деталей на каждом уровне;
3. Выделение и привлечение внимание к существенным частям проблемы;
4. Применение строгого систематического подхода к решению проблемы;
5. Обоснование и совместимость элементов;
6. Структурированность и организованность данных.

При проектировании программного обеспечения автоматизированной информационной системы, созданные модели расширяются, уточняются и дополняются различными диаграммами, отражающие структуру данного программного обеспечения. В них входят: архитектурная модель программного обеспечения, структурные схемы диаграмм и программных форм. Все перечисленные модели дают полное представление о проектируемой информационной модели.

2.1.2 Модель информационной системы управления рисками нагонных наводнений в городе Санкт-Петербург

На протяжении долгих лет человечество пытается найти способы защиты от наводнений, но безрезультатно: с каждым годом наводнения наносят все больший ущерб. Во всем мире, включая Россию, наблюдается тенденция значительного роста ущербов от наводнений, вызванная нерациональным ведением хозяйства, усилением их хозяйственного освоения и потеплением климата.

Причиной наводнений в реках Заполярья являются ветровые нагоны.

Физический механизм формирования нагонов и сгонов – это воздействие ветра и атмосферного давления на водную поверхность, которое вызывает перераспределение водных масс внутри бассейна. Колебания атмосферного давления обычно вызывают изменения уровня воды - повышения в областях пониженного атмосферного давления и понижения в антициклонах. Изменения уровня воды возбуждают длинные волны. Возникающие движения, таким образом, являются результатом совместного воздействия длинных внешних волн (в случаях бассейнов с открытой границей), локальных ветров (особенно в мелководных районах) и градиентов атмосферного давления (в глубоководных районах). Основной характеристикой, по которой можно судить о величине нагона, является нагонный подъем уровня воды, обычно выражающийся в метрах. Другими величинами служат глубина распространения нагонной волны, площадь и продолжительность затопления. На величину нагонного уровня влияют скорость и направление ветра. Нагонные наводнения нередко охватывают большие территории. Продолжительность затопления обычно находится в пределах от нескольких часов до нескольких суток. Чем крупнее водоем и меньше его глубина, тем больших размеров достигают нагоны. [42]

Причиной наводнений в Санкт-Петербурге являются циклоны в Балтийском море с обилием западных ветров, которые и производят нагонную волну,двигающуюся в направлении устья Невы, где она вступает в «конфликт» с природным течением реки.

Подъемы воды усиливаются из-за мелководья и пологости дна в Невской губе, а также из-за значительного сужения к дельте реки.

Наводнения возникают и по другим причинам: появляющиеся циклоны на Балтике с обилием ветров западного направления вызывают подъем «медленной» нагонной волны Кельвина и её движение в направлении устья реки, где она встречается с противоположным направлением естественного течения Невы.

Продолжительность наводнения в Реке Нева находится в пределах от 8 часов до 2 или 3 дней и в среднем составляют 24 часа, поэтому так важно знать заранее возможность наводнения.

Впервые предупреждения об угрозе наводнения в Петербурге стали составляться Главной Физической обсерваторией в 1897 году. До этого оповещение о уже наступившем наводнении производились выстрелами из пушки, барабанным боем и вывешиванием флагов на Адмиралтействе. Разрушительное наводнение 1924 года стало причиной появления исследований В.Ю.Визе, С.А. Соватова, В.М. Манеева, В.А.Берга и других. Во всех этих работах рассмотрена волновая гипотеза с той лишь разницей, что одни считали ветер главной причиной, а другие волне уделяли больше внимания. Но ни одна из этих работ практического применения не получила. Правда в 1936 году К.П. Турыгин составил типизацию наводненческих циклонов и создал Атлас Невских наводнений, который почти в течении 20 лет являлся основным пособием синоптиков для составления штормовых предупреждений о наводнениях. В 1954 году Н.И.Бельским впервые в истории службы предупреждений создан эмпирический метод расчёта высоты подъёма уровня воды в устье реки, который до сих пор используется для прогнозирования. Но этот метод позволяет спрогнозировать максимальную высоту подъёма уровня воды с заблаговременностью 5-8 часов. Всем понятно, что этого времени недостаточно для принятия мер по снижению размера ущерба в условиях огромного города.

Была ещё одна попытка создания метода прогноза Невских наводнений с заблаговременностью 12 часов и более. Но в связи с низкой оправдываемостью прогнозов нигде не применяется. Так что разработка методов, дающих возможность рассчитать подъёмы уровня воды в устье реки Нева с большей заблаговременностью, остаётся одной из важнейших задач.

Сегодня наука достаточно далеко продвинулась в разработке технологий прогнозирования. Наиболее существенное место занимают теория вероятностей и математическая статистика, теория численных методов анализа и оптимизации, современная теория факторного анализа, дифференциальные уравнения.

В основном современные прогнозные модели объектов строятся в рамках статистических моделей, моделях экстраполяции и интерполяции регулярных составляющих, оценки влияния случайных составляющих процесса.

Наряду с вышеизложенными моделями можно предложить стохастические модели прогноза состояния водной среды, основанные на состоянии параметров технических систем. Это могут быть одно- или многопараметрические модели, созданные на базе Марковских процессов, позволяющих прогнозировать вероятность невыхода параметров из заданной зоны. Ведь, по сути, по двум точкам (замерам уровня воды у Горного института и Кронштадтского

футштока) можно построить двумерную модель, описывающую случайный процесс нахождения в заданный нами период реализации уровень водной поверхности реки Нева. В дальнейшем при применении математических и статистических методов, исследователь, задавая свои границы, при которых, по его мнению, происходит наводнение, должен рассчитать вероятность выхода уровня воды за эти границы, то есть возможность наводнения. Важным аспектом данной модели является дальнейшее исследование этих процессов с помощью компьютерного моделирования. [43]

На основании предложенной модели было проведено компьютерное моделирование стохастического процесса, имитирующей поведение водной поверхности, создана информационная система, в которой можно рассчитать вероятность возникновения наводнения.

При создании данной системы было разработано несколько моделей таких, как математические, имитационные и компьютерные модели.

Математическая модель описывала статистические характеристики процессов и использовала следующие формулы [43]:

$$r_{x,y}(t) = e^{-bt}. \quad (2.1.2.1)$$

, где автокорреляционная функция исследуемого процесса $r_{x,y}(t)$ описана в виде экспоненциальной функции;

$$\rho_{x,y}(\tau) = \text{const}. \quad (2.1.2.2)$$

, где расчёт взаимной корреляционной функции $\rho_{x,y}(\tau)$ производится при условии не изменчивости её во времени;

$$P^*(t_3/x_0) = \Phi(B) - \Phi(A). \quad (2.1.2.3)$$

$$\text{Где: } B = \frac{b - m_x - r_x(t_3) \cdot (x_0 - m_x)}{\sigma_x \sqrt{1 - r_x^2(t_3)}},$$

$$A = \frac{a - m_x - r_x(t_3) \cdot (x_0 - m_x)}{\sigma_x \sqrt{1 - r_x^2(t_3)}},$$

$$\Phi(Z) = \frac{1}{\sqrt{2\pi}} \cdot \int_{-\infty}^Z e^{-\frac{x^2}{2}} dx,$$

$r_x(t_3) = e^{\beta t_3}$ - нормированная автокорреляционная функция случайного процесса.

$$P_*(t_3/x_0) = 1 - \frac{\int_0^{e_3} |W(t/x_0)| dt}{\sqrt{2\pi \cdot (1 - r_x^2(t_3))}} \quad (2.1.2.4)$$

$$\Gamma \text{де: } W\left(\frac{t}{x_0}\right) = [y'(t) + p'(t) \cdot y(t)] \cdot e^{-\frac{1}{2}y^2(t)} - [\mu'(t) + p'(t) \cdot \mu(t)] \cdot e^{-\frac{1}{2}\mu^2(t)}$$

$$y(t) = \delta - \frac{r_x(t_3) \cdot [x_0 - m_x]}{\sigma_x}; y'(t) \equiv 0;$$

$$\mu(t) = \varepsilon - \frac{r_x(t_3) \cdot [x_0 - m_x]}{\sigma_x}; \mu'(t) \equiv 0;$$

$$\rho(t) = \frac{1}{r_x(t)} = e^{\beta \cdot t};$$

$$\rho'(t) = \frac{\rho(t)}{dt} = \beta \cdot e^{\beta \cdot t};$$

$$\delta = \frac{a - m_x}{\sigma_x};$$

$$\varepsilon = \frac{b - m_x}{\sigma_x}$$

, где расчет вероятностных характеристик процессов $(P^*(\frac{t_3}{x_0}))$ -верхняя оценка вероятности невыхода процесса за пределы заданных границ и за период времени от t_0 до t_3 , $P_*(\frac{t_3}{x_0})$ - нижняя оценка вероятности невыхода процесса за пределы заданных границ и за период времени от t_0 до t_3) осуществляется при условии, что данные процессы являются стационарными, нормальными процессами.

Отсюда следует, что оценки верхней и нижней границы вероятности наводнения будут:

$$1. \text{ верхняя оценка: } Q^* = 1 - P_*(\frac{t_3}{x_0}); \quad (2.1.2.5)$$

$$2. \text{ нижняя оценка: } Q_* = 1 - P^*(\frac{t_3}{x_0}). \quad (2.1.2.6)$$

Компьютерное же моделирование предусматривает создание информационной модели и модели данных, что и было сделано при разработке информационной системы прогнозирования рисков нагонных наводнений в городе Санкт-Петербург (Рисунки 2.1.2.1 – 2.1.2.2).

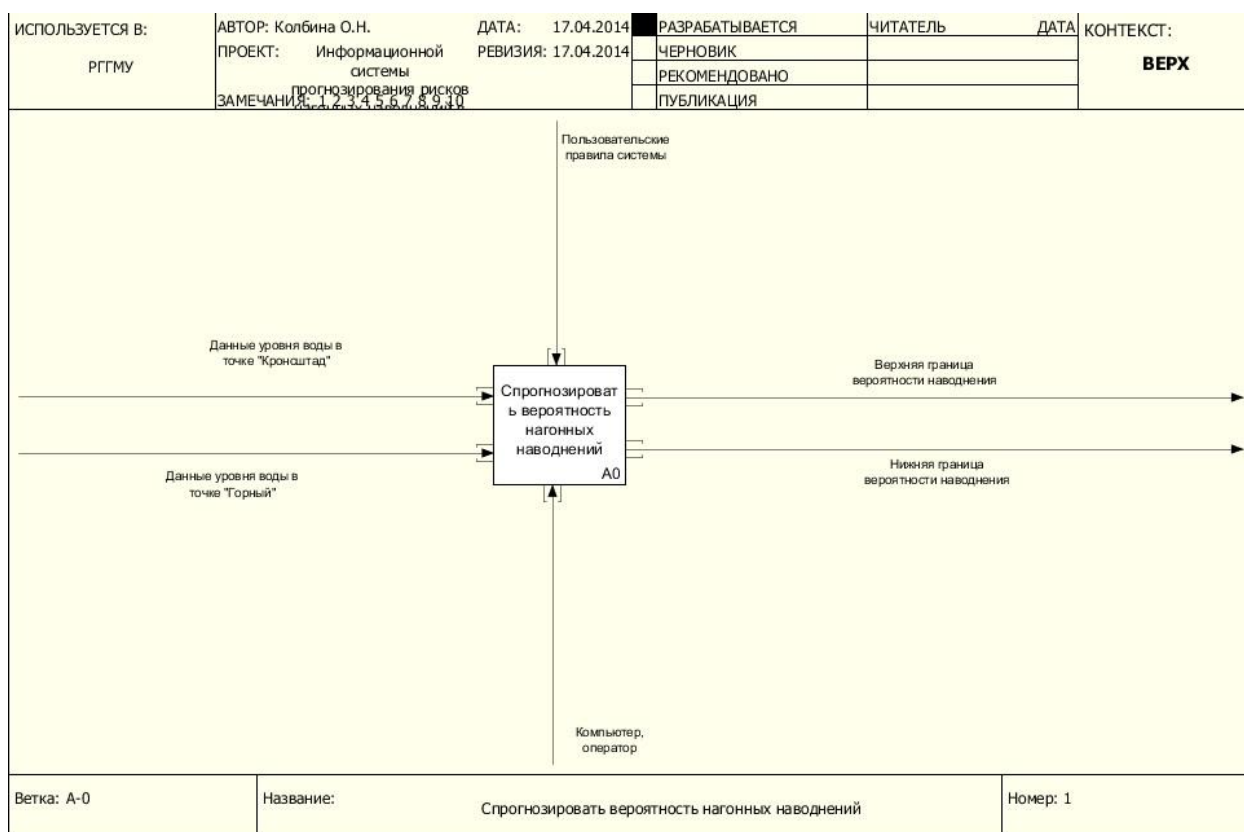


Рисунок 2.1.2.1 Контекстная диаграмма функциональной модели системы прогнозирования рисков нагонных наводнений

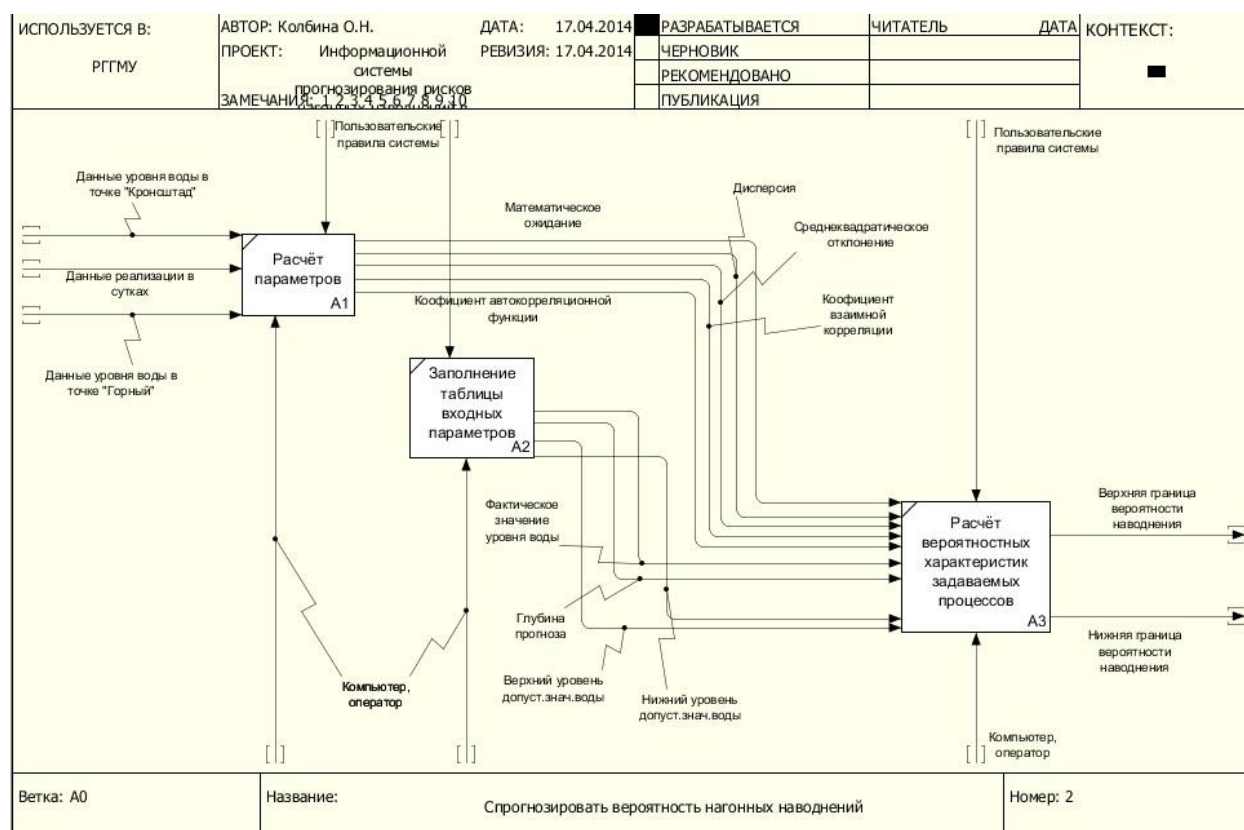


Рисунок 2.1.2.2 Диаграмма декомпозиции системы прогнозирования рисков нагонных наводнений

Данная модель была создана при использовании программы Ramus, которая позволяет проектировать DFD и IDEF0 диаграммы, отвечающие за моделирование всей системы как набора чередующихся функций.

2.2 Разработка модели геоинформационной системы на основе гетерогенных баз данных

Создание механизмов управления распределенной информационной системой на основе гетерогенных баз данных требует соблюдение некоторых ограничений, а именно – учета гетерогенности, автономности и распределенности системы баз данных.

За несколько последних лет было предложено немало технологий борющихся с гетерогенностью и распределенностью базы данных (CORBA, JAVA, EJB), но ни одна из них не может обеспечить сохранение автономности источников данных. Поэтому на данный момент широкую известность получили лишь подходы доступа к гетерогенной базе данных распределенной информационной системы только для чтения данных. Такое состояние вызвано простотой реализации этого подхода, так как операции чтения двух независимых транзакций не конфликтуют и, соответственно, не могут нарушить целостности и непротиворечивости данных в распределенной информационной системе. [49] Развитие технологий и требования пользователей подталкивает разработчиков к созданию такой структуры информационной системы, где предусматривалась бы возможность доступа пользователей как локальных, так и глобальных, не только для чтения данных, но и для их изменения.

Рассмотрим теоретический аппарат управления модифицирующими глобальными транзакциями, базирующийся на контролировании допуска глобальных транзакций в соответствии с их временными метками. [49]

Любое взаимодействие информационной системы с базой данных осуществляется по средствам транзакций. Гетерогенные базы даны не исключение.

Транзакцией называется единая логическая единица работы, все изменения которой целиком либо сохраняются, либо откатываются. [48] В информационной системе будут использоваться модель плоских транзакций. Она так называется в связи с её атомарностью. Структура такой транзакции состоит из начального этапа транзакции, последующих операций и конечного этапа.

Транзакции в распределенных информационных системах должны обеспечивать конкурентное выполнение именно глобальных транзакций, так как именно это является

гарантией целостности и непротиворечивости данных во всей системе. Но существует также вероятность возникновения распределенных тупиков, которые возникают при применении блокирующего характера механизмов конкурентного контроля.

Если рассматривать теорию управления транзакциями, то в ней существует два параллельных направления – сериализуемость [45] и надежность [44].

Сериализуемость заключается в корректности выполнения алгоритмов управления транзакциями и применима в программной среде без отказов. Надежность, в свою очередь, также является критерием корректности, но в программной среде с возможными отказами. Надежность можно описать используя такие свойства транзакций как непротиворечивость и сохранность данных, так как отказы могут приводить к потере данных. Поэтому распределенной информационной системе нужно реагировать на такие ситуации автоматически. Обеспечение надежности в РИС несёт в себе ряд особенностей, по сравнению с СУБД. В ней приходится выбирать между сохранением локальной автономии и использованием механизма эффективного глобального восстановления. Для организации транзакций в информационной системе с возможностью отказов необходимы два вида протокола. Первый вид протокола, который называется протокол атомарной фиксации транзакций, необходим для завершения глобальной транзакции при одинаковом состоянии на всех узлах системы. Вторым протоколом, называется протокол восстановления, которым определяются действия в системе после её отказов.

Выделяют следующие основные направления управления транзакциями в распределенной информационной системе на основе гетерогенных баз данных:

1. Назначение критерием корректности выполнения транзакций глобальную сериализуемость;
2. Поиск других критериев корректности по средствам снижения требований к системе;
3. Разработка новых расширенных моделей транзакций.

Критерий корректности указывает на допустимость выполнения конкурентных транзакций в информационной системе. А глобальная сериализуемость, которая предполагает возможность управления всеми данными, не предусматривается в гетерогенной среде баз данных. Поскольку автономность компонентов системы предполагает отсутствие любого глобального управления, а также любых записей о выполнении этой транзакции в локальных СУБД.

В однотипных распределенных базах данных глобальная сериализуемость выполняется по средствам схемы конкурентного выполнения, которую использует каждый узел при выполнении глобальной транзакции. При этом многие созданные системы и модели,

рассчитанные на выполнение автономных операций не подходят для организации систем с гетерогенными базами данных. Поскольку они не предоставляют информацию и сервис, которые необходимы для проведения глобальных транзакций.

Глобальная сериализуемость достигается при следующих условиях [46, 47]:

1. История выполнения транзакций в каждой гетерогенной базе данных локально сериализуема;
2. Относительный сериализационный порядок глобальных подтранзакций тот же самый в каждой гетерогенной базе данных.

Однако существует возможность косвенных конфликтов между глобальными транзакциями, которые не известны на глобальном уровне распределенной информационной системы. Существуют ситуации, при которых даже последовательное выполнение транзакций не может обеспечить глобальной сериализуемости (Рисунок 2.2.1).

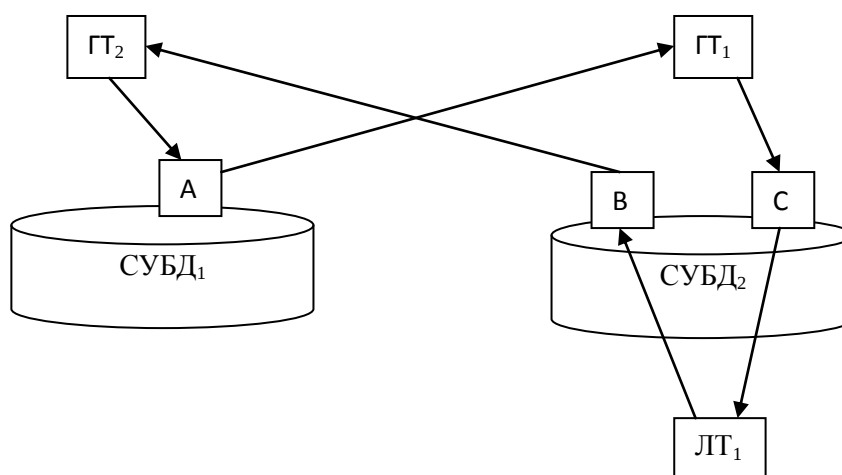


Рисунок 2.2.1 Невыполнение глобальной сериализуемости в распределенных информационных системах

В основе распределенной системы выполняется две глобальные транзакции GT_1 и GT_2 , и одна локальная транзакция LT_1 . В рассматриваемом примере в двух локальных СУБД имеются подтранзакции у двух глобальных транзакций. В СУБД₁ глобальная транзакция GT_1 читает A и глобальная транзакция GT_2 потом производит операцию записи в A. Записать локальные операции транзакций в СУБД₁ можно следующим образом:

$$R_{GT_1}(A)W_{GT_2}(A) \quad (2.2.1)$$

, где $R_{GT_1}(A)$ – это функция чтения из объекта данных A,

$W_{GT_2}(A)$ – это функция записи в объект A.

Отсюда следует, что транзакции GT_1 и GT_2 прямо конфликтуют в СУБД₁. Сериализационный порядок выполнения транзакций в этой СУБД можно представить в виде зависимости второй глобальной транзакции от первой глобальной транзакции (Рисунок 2.2.2).

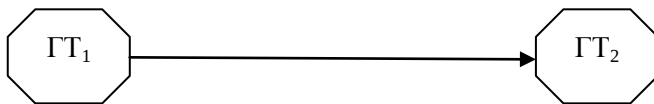


Рисунок 2.2.2 Зависимость транзакций в СУБД₁

В СУБД₂ глобальные транзакции GT₁ и GT₂ могут обратиться к различным объектам данных. Так глобальная транзакция GT₁ записывает С, а позднее GT₂ читает В. Запись локальных операций транзакций в СУБД₂ можно следующим образом:

$$R_{LT_1}(C) W_{GT_1}(C) R_{GT_2}(B) W_{LT_1}(B) \quad (2.2.2)$$

Прямого конфликта в СУБД₂ нет, но так как локальная транзакция LT₁ записывает В и читает С, то глобальные транзакции между собой косвенно конфликтуют. Конфликт происходит из наличия локальной транзакции. Сериализационный порядок выполнения транзакций в СУБД₂ можно описать зависимостью первой глобальной транзакции от выполнения локальной транзакции, которая в свою очередь будет зависеть от выполнения операций второй главной транзакции (Рисунок 2.2.3).

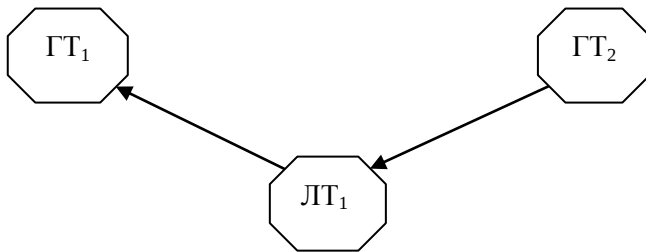


Рисунок 2.2.3 Зависимость транзакций в СУБД₂

Поскольку гетерогенная база данных автономна, распределенная информационная система имеет контроль только над глобальными транзакциями, а о локальных транзакциях и косвенных конфликтах, которые могут возникнуть, ей информация не доступна. Система может определить лишь прямые конфликты между глобальными транзакциями, такими как GT₁ и GT₂ в СУБД₁.

Основными трудностями при обеспечении глобальной сериализуемости в распределенной информационной системе на основе гетерогенных БД являются предотвращение косвенных конфликтов и возникновение глобальных тупиков.

Глобальные тупики возникают при применении системой гетерогенных баз данных локальных механизмов управления транзакциями на основе блокирования объектов данных. Условия при которых происходит тупик заключаются в существовании множества глобальных транзакций GT₁...GT_n, и множества гетерогенных баз данных ГБД=∪_jГБД_j, с которыми происходят следующие ситуации ожидания [49]:

1. GT₂ ожидает завершения транзакции GT₁ в ГБД_i, отсюда следует, что GT_{i+1} ожидает GT_i в ГБД_i, где 1<i<n, а GT₁ ожидает GT_n в ГБД_n;

2. подтранзакции $ГПТ_1^j$ глобальной транзакции $ГТ_1$, которые выполняются в гетерогенной базе данных $ГБД_j$ при $j=1..n$, находятся в состоянии готовности к фиксации, но не могут быть зафиксированы пока оставшиеся подтранзакции оставшихся глобальных транзакций не находятся в таком же состоянии;
3. подтранзакция, которая находится в состоянии готовности к фиксации, не освобождает ресурсы пока не зафиксируется или откатится.

Поскольку локальные гетерогенные базы данных не обмениваются управляющей информацией, возникают тупиковые ситуации (Рисунок 2.2.4). Соответственно распределенная информационная система не имея информации о локальных транзакциях также не определит тупиковую ситуацию, что может привести к сбою в работе всей системы.

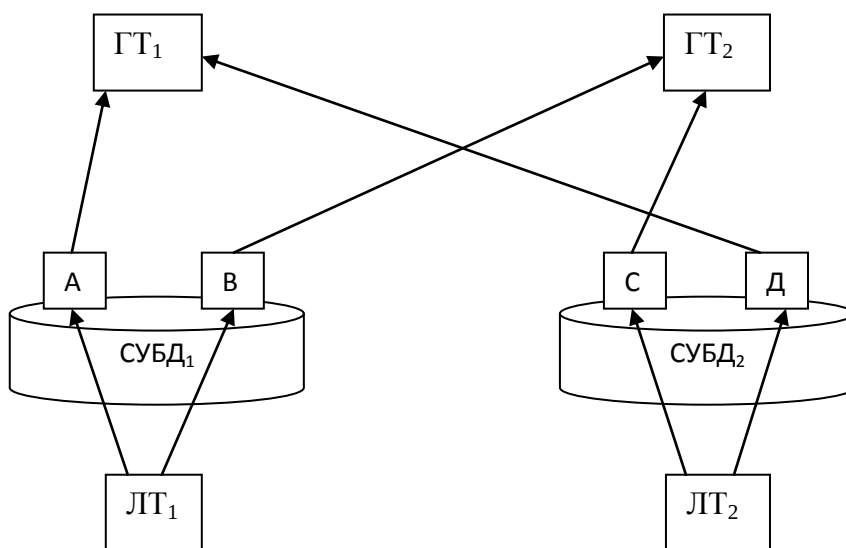


Рисунок 2.2.4 Недопуск возникновения тупиковых ситуаций в распределенных информационных системах

В распределенной информационной системе над объектами данных А, В, С, Д выполняются две глобальные транзакции $ГТ_1$, $ГТ_2$ и две локальные транзакции $ЛТ_1$, $ЛТ_2$. Пусть глобальная транзакция $ГТ_1$ выполнит операцию $R_{ГТ1}(A)$, а транзакция $ГТ_2$ выполнит операцию $R_{ГТ2}(C)$. После чего в гетерогенной базе данных с СУБД₁ локальная транзакция $ЛТ_1$ выполняет операцию $W_{ЛТ1}(B)$, допускает $W_{ЛТ1}(A)$ и переходит в состояние ожидания снятия блокировки с объекта А транзакцией $ГТ_1$. В СУБД₂ локальная транзакция $ЛТ_2$ выполняет операцию $W_{ЛТ2}(D)$ и допускает выполнение операции $W_{ЛТ2}(C)$. После чего переходит в состояние ожидания снятия блокировки с объекта С транзакцией $ГТ_2$. В завершении всех операций, транзакции $ГТ_1$ и $ГТ_2$ допускают выполнение оставшихся незавершенными операций и возникает глобальный тупик.

Выделяют два основных способа борьбы с взаимной блокировкой транзакций [50]:

1. предупреждение ошибочных ситуаций;
2. выявление взаимных блокировок с последующим их устранением.

Предупреждающий метод заранее определяет возможные ситуации, при которых возникает блокировка транзакций, и предотвращает их. Выявление же взаимных блокировок с последующим их устранением подразумевает появление подобных ситуаций в системе, которые впоследствии, при их обнаружении, ликвидируются. Наиболее подходящим вариантом для распределенной информационной системы на основе гетерогенных баз данных является первый подход. Это связано со сложностью выявления и устранения возможных тупиковых ситуаций, которые подразумевают отмену одной или нескольких транзакций.

Созданием универсального протокола, создаваемого полностью сериализуемое управление транзакциями занимались многие ученые с начала 90-х годов. Одним из них был Брейтбард (Breitbart), который предложил алгоритм управления локальными гетерогенными базами данных, при использовании ими в качестве механизма конкурентного выполнения транзакций строгий двухфазный протокол блокирования (strict 2PL). Согласно этому протоколу глобальная транзакция не фиксируется, пока все операции чтения записи этой транзакции не завершатся на всех узлах системы. [40] Этого вполне достаточно для обеспечения глобальной сериализуемости, но данный алгоритм вызывает недостаточную конкурентоспособность выполнения транзакций. Что возникает из-за блокировки глобальной транзакцией на всех узлах распределенной системы операций, пока выполнение не завершится на всех узлах. Алгоритм особенно не применим к транзакциям, проходящим через многие узлы системы и долгоживущим транзакциям. Его использование может привести к возникновению распределенных тупиков.

Георгакополос (Georgakopoulos) предложил подход, основанный на использовании временных меток в каждой локальной гетерогенной базе данных. Данный алгоритм было решено назвать ОТМ (Optimistic Ticket Method). Его действие направлено на установку временного интервала для глобальной транзакции и отправку подтранзакции на выполнение действия. После того как подтранзакции перешли в состояние готовности к фиксации происходит анализ сериализационного графа на циклы. Глобальная транзакция фиксируется только если она не вовлечена в тупиковую ситуацию или не произошло отката подтранзакции. Однако при большом количестве глобальных транзакций, которые затрагивают различные узлы распределенной информационной системы, приводят к множеству конфликтов глобальных транзакций и к последующему их откату, что приводит к перегрузке системы гетерогенных баз данных.

Не мене важным в системе баз данных являются механизмы обеспечения надежности. Надежность заключается в понятии устойчивости системы управления базами данных к различным типам отказов, а также к способности восстановления после их появления.

Наиболее распространен метод борьбы с отказами по средствам журнала, в который записываются все выполняемые операции транзакций, и по которому можно восстановить состояние всей распределенной системы. Обычно записываются только те транзакции, которые выполнены, а те, которые откатились, в журнал не заносятся. В случаях распределенной информационной системы без отказов используют механизм атомарной фиксации транзакций. [52] Он заключается в использовании модифицированных протоколов, которые фиксируют подтранзакции глобальной транзакции до фиксации самой глобальной транзакции. Однако этот подход содержит противоречия, так как если глобальная транзакция откатывается, то необходимо устранение уже зафиксированных подтранзакций.

Выделим достаточные условия для безотказного управления транзакциями в распределенной информационной системе на основе гетерогенных баз данных:

1. все гетерогенные базы данных должны быть локально атомарны;
2. если все операции множества глобальных транзакций $МГТ_i$ выполняется перед множеством $МГТ_j$ в какой-либо отдельно взятой СУБД, тогда то же самое должно выполняться для всех остальных подтранзакций;
3. в отдельно взятой гетерогенной базе данных не должно выполняться больше одной подтранзакции глобальной транзакции;
4. распределенная информационная система на основе гетерогенных баз данных должна справляться с появлением глобальных тупиков.

В распределенной информационной системе на основе гетерогенных баз данных должно существовать условие автономности интегрируемых баз данных. Для сохранения этого условия необходим протокол управления модифицирующими глобальными транзакциями. В качестве критерия корректности будет выбрана глобальная сериализуемость. Для такого протокола должны быть характерны следующие признаки:

1. каждая локальная СУБД должна гарантировать локальную сериализуемость и безкаскадные истории;
2. локальные истории выполнения транзакций должны быть без тупиков.

Использование данного протокола управления транзакциями предусматривает определенную модель управления распределенной информационной системой на основе гетерогенных баз данных, обеспечивающую интеграцию разрозненных баз данных в единое информационное пространство (Рисунок 2.2.5).

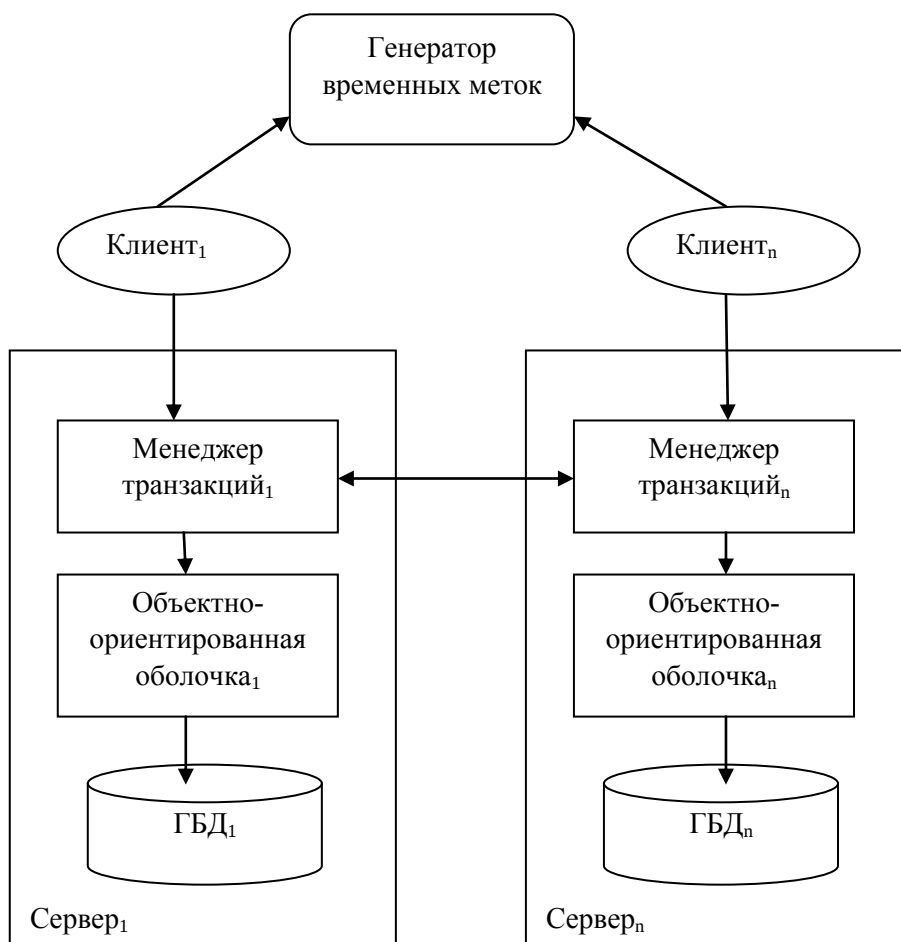


Рисунок 2.2.5 Модель системы управления распределенной информационной системы на основе гетерогенных баз данных

На представленном рисунке рассмотрена архитектура системы, основанная на отдельно взятой гетерогенной базе данных, к каждой из которых можно обратиться по средствам одних и тех же методов доступа.

Модель должно успешно работать в многопользовательском режиме и предотвращать рассогласование данных, вызванное одновременной работой с одними и теми же данными, а так же организовывать программную архитектуру.

Структурная модель геоинформационной распределенной системы на базе гетерогенных БД должна быть основана в форме многоуровневой архитектуры (Рисунок 2.2.6).

Клиент-серверная модель соединений, взявшая за основу принципы структурного программирования, а именно разделения процессов ввода-вывода и вычислительной обработки, в последнее время не справляется с растущими потребностями разработчиков в области надежности и производительности систем.

Наиболее отвечающая всем требованиям равномерного распределения вычислительной нагрузки, обеспечения надежности и масштабируемости, является многоуровневая модель создания сложных программных систем. Она содержит три уровня представления данных

«данные – обработка – интерфейс». Достоинствами данного представления являются независимость системы от количества программных компонентов, входящих в каждый уровень, свой набор протоколов и прикладных интерфейсов для каждого уровня, определяющий свою архитектуру.

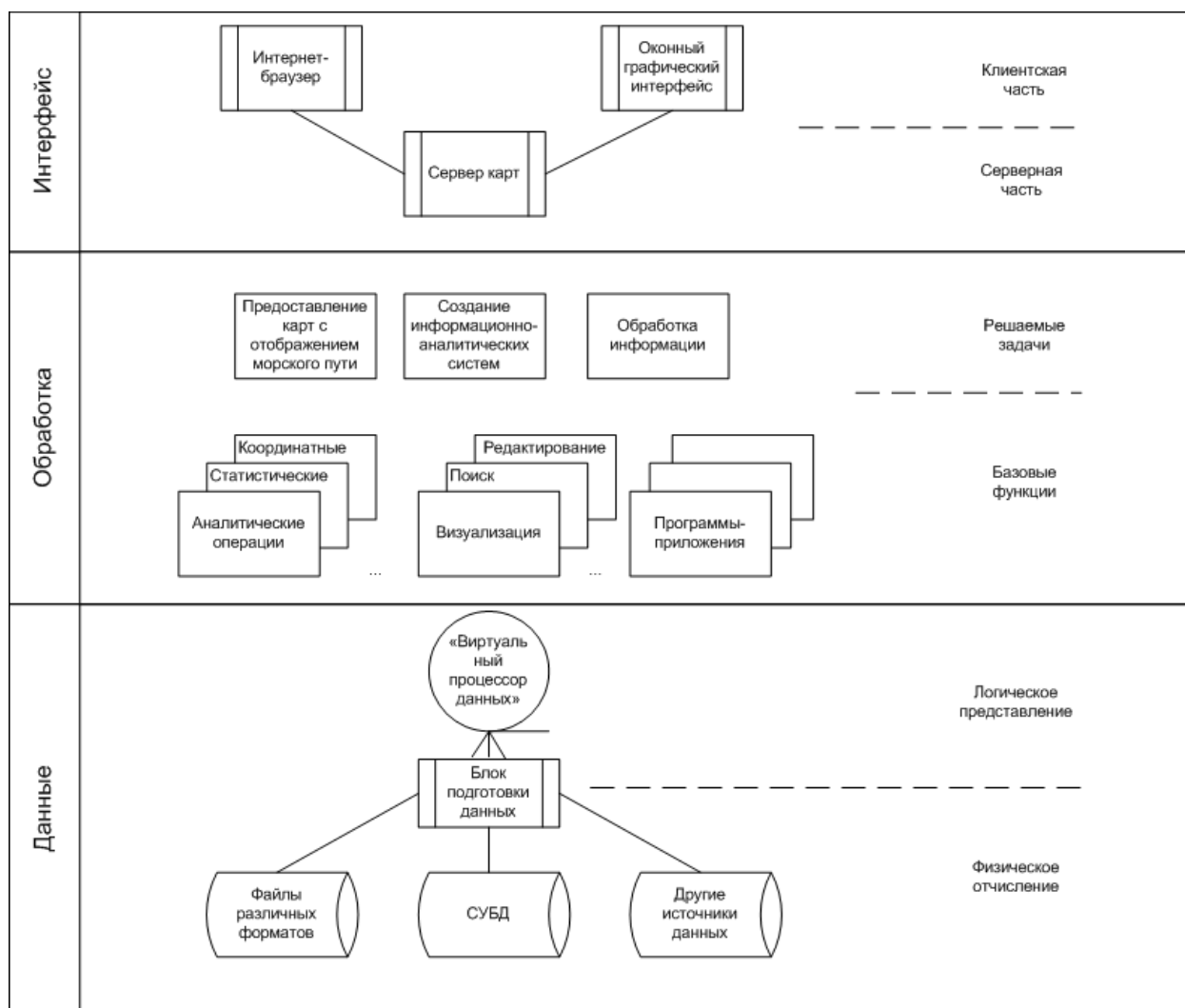


Рисунок 2.2.6 Структурная модель геоинформационной распределенной системы на основе гетерогенных баз данных

На основании структурной модели формируется потоковая модель, которая представляет структуру прохождения информационных данных в рамках решаемой задачи (рисунок 2.2.7).

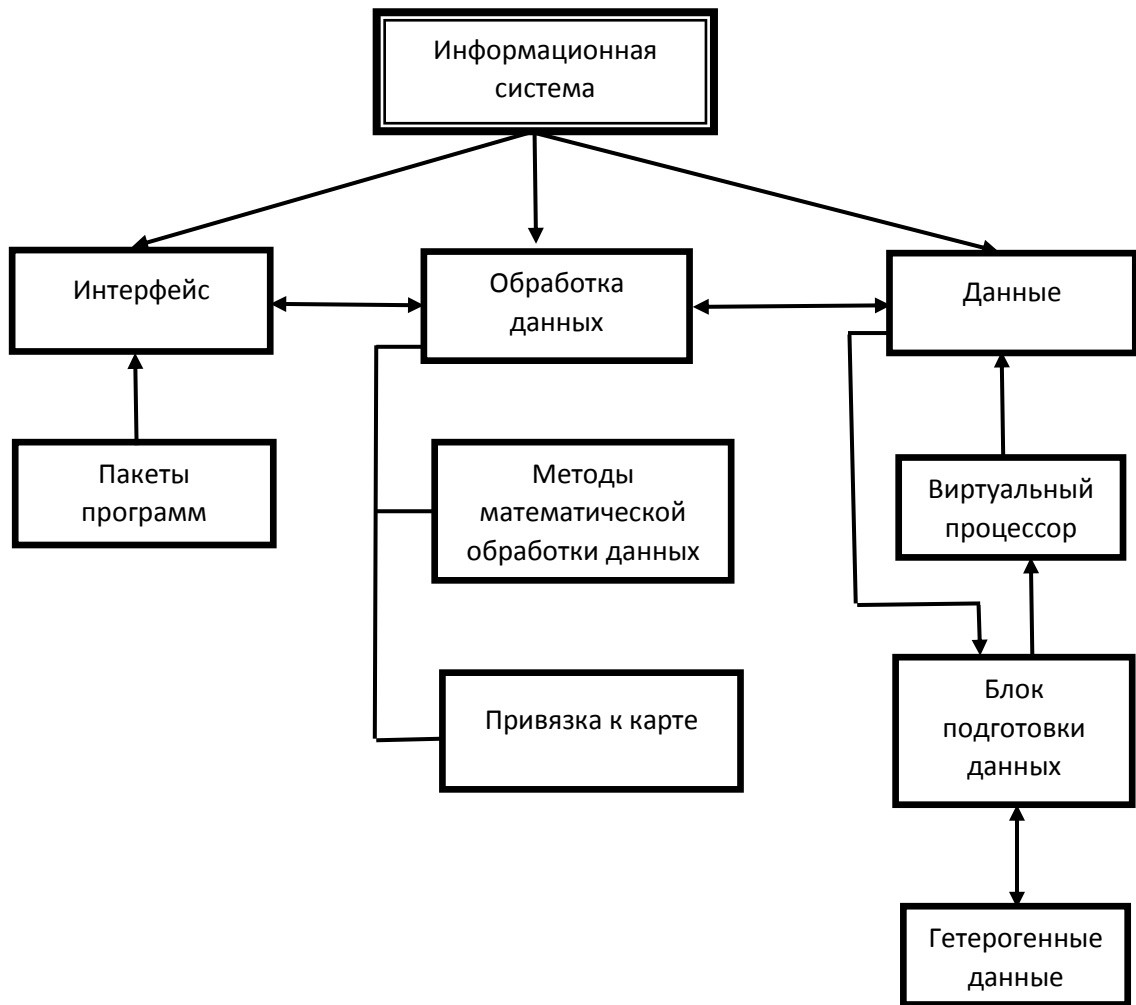


Рисунок 2.2.7 Поточковая модель геоинформационной распределенной системы на основе гетерогенных баз данных

Традиционно организация данных для их совместного использования представляла собой создание централизованного хранилища. Однако организация такой базы данных на начальных этапах создания распределенной геоинформационной системы представляется не целесообразной, поскольку преобразование всех необходимых данных займёт много времени и есть вероятность потери части информации, не говоря уже о работе локальных баз данных и их обновлении.

Во избежание трудностей доступа к данным принято решение использовать механизм «виртуального процессора данных». Здесь он определяется как программно-технологический модуль, который позволяет единообразно получить доступ ко всем данным системы через стандартные интерфейсы (протоколы) и скрывающий при этом особенности размещения данных, структуры и форматы данных. При этом нет необходимости преобразования данных к формату системы, они остаются в исходном представлении и могут использоваться локально своими программными продуктами.

При более детальном рассмотрении «виртуального процессора данных» можно утверждать, что он состоит из набора не конкретизированных виртуальных баз данных. Некий виртуальный банк данных. Информационными источниками баз фактических состояний является фактический цикл функционирования организационных подсистем сложного объекта управления, при рассмотрении геоинформационной системы как системы принятия управленческих решений. Информационными источниками баз плановых состояний и нормативных состояний могут являться входные документы, виртуальные результаты принятия управленческих решений, а также результаты решения регламентированных задач. В соответствии с регламентированными задачами и ситуациями в соответствующих базах гетерогенного банка данных текущего периода, а также, при необходимости, из баз предшествующих периодов, целесообразно формировать локальные по отношению к процессору базы данных.

Основным критерием распределения этих баз будет локализация по месту решения соответствующей задачи или месту управления соответствующей ситуацией.

В свою очередь, для решения задач управления Арктическими территориями РФ или в нерегламентированных ситуациях управления необходимо на основе виртуальной базы данных, гетерогенных баз данных и результатов обработки ситуаций формировать персональные базы для индивидуального пользователя данной системой. Соответственно, местоположение лица, принимающего решение будет критерием формирования персональной базы и обращения к ближайшему механизму «виртуального процессора данных».

После принятия пользователем соответствующих действий по решению необходимых задач и формирования запросов на предоставление конкретных данных, из разных локальных баз данные определяются на рассогласованность. Если таковая существует, формируются базы рассогласованных состояний. В основном в них попадают данные из гетерогенных локальных баз данных, после чего они проходят сложные процедуры согласования и попадают в базы согласованных состояний, где могут находиться данные из идентичных, в основном реляционных баз. При этом следует отметить, что виртуальные персональные базы составляют основу геоинформационной системы управления территориями.

Из баз согласованных состояний данные попадают в соответствующие базы плановых, нормативных или вероятностных состояний.

Кроме различных баз состояний необходимо ввести в механизм «виртуального процессора данных» интегрированную базу данных, где воедино сливаются и преобразуются данные об исследуемых объектах. При таком подходе сохраняется возможность использования единой технологии ведения данных.

Следует отметить, что все процедуры контроля и преобразования данных в подсистеме данных, т.е. процедуры технологии динамической организации и реорганизации подсистемы данных, выполняются интеллектуальной системой анализа, контроля и преобразования информации. Эта система выполняет преобразования на основе знаний, представленных в форме продукций. Ее можно рассматривать как надстройку над традиционными системам управления данными.

Все вышеприведенные рассуждения особенно важны, когда возникает необходимость рассматривать системы управления как некоторое множество взаимосвязанных подсистем в виде сети, имеющих фиксированные сферы компетенции в области принятия решений всей системы в целом. В этих условиях становится актуальной задача обеспечения координации и взаимодействия при решении отдельных подзадач, решаемых в подсистемах сети. Централизованное решение таких задач на практике, как правило, не выгодно или просто невозможно в силу их огромной размерности физической распределенности, что приводит к необходимости разработки специальных распределенных методов их решения.

Схема алгоритмов согласования решений рассмотрена в работе доктора технических наук Истомина Евгения Петровича. [53]

Набор же драйверов в «блоке подготовки данных» для «виртуального процессора данных» и других его компонентов могут при необходимости расширяться без каких-либо существенных ограничений и проблем. Самое главное в дополнениях придерживаться обеспечения преемственности программных интерфейсов (протоколов).

2.3 Модификация методики проектирования геоинформационной системы на основе распределенных гетерогенных баз данных

В двадцать первом веке информационные технологии могут предоставить широкий набор способов создания информационной системы. Выбор происходит исходя из требований предполагаемых пользователей, которые могут измениться несколько раз в процессе разработки.

В качестве проекта информационно-управляющей системы обозначим проектно-конструкторскую и технологическую документацию, в которой описаны проектные решения, используемые при реализации системы.

Процесс проектирования информационной системы означает преобразование входной информации об объекте, методах и опыте проектирования в проект ИС. В соответствии с

государственными стандартами 34-й и 19-ой серии проектирование информационной системы сводиться к следующим стадиям [54, 55, 56, 57, 58, 59]:

1. Формирование требований к автоматизируемой системе;
2. Разработка концепции автоматизируемой системы;
3. Техническое задание;
4. Эскизный проект;
5. Технический проект;
6. Рабочая документация;
7. Ввод в действие;
8. Сопровождение автоматизируемой системы.

Прежде чем начать проектирование информационной системы необходимо определиться с объектом проектирования.

Под объектом проектирования понимают отдельные элементы или комплексы функциональных и обеспечивающих частей. Функциональными элементами принято считать задачи или комплексы задач и функции управления. Обеспечивающая часть состоит из элементов или комплексов информационного, программного, технического и других видов обеспечения системы [60].

Для геоинформационной системы управления территориями Заполярья объектом проектирования служат метеорологические, гидрологические и иные параметры оказывающие влияние на хозяйственную деятельность в Арктике.

Помимо объекта проектирования существует ещё и субъект проектирования информационной системы, который может состоять из коллективов специалистов, осуществляющих проектную деятельность, и заказчика требуемой ИС.

Субъектом проектирования в данной системе являются муниципальные органы власти в районах Заполярья РФ, а также крупные промышленные компании, заинтересованные в проведении подобных исследованиях.

Масштаб проектируемой системы определяет количество проектного коллектива и время, отведенное на создание автоматизированной системы. В случае создания геоинформационной системы управления арктическими территориями на основе распределенных гетерогенных баз данных, существует необходимость в участии нескольких проектных коллективов на местах распределения баз данных и создании головной организации, которая бы координировала деятельность команд-исполнителей на местах.

Проектирование информационной системы предусматривает использование разработчиками определенной технологии проектирования, которую можно охарактеризовать как совокупность методологии и средств проектирования ИС, а также методов и средств его

организации. В её основе лежит технологический процесс, определяющий действия, их последовательность, а также требуемый состав исполнителей, средств и ресурсов.

Проектирование информационной системы можно разделить на ряд взаимосвязанных, последовательно или параллельно выполняющихся, цепочек действий. Они, в свою очередь, делятся на проектировочные операции, которые формируют или редактируют результаты проектирования. Таким образом технология проектирования выполняется на основе того или иного метода по средствам выполнения последовательных операций, в результате чего становится понятным не только что должно быть сделано, но и как, кем и в какой последовательности.

Основу же технологии проектирования составляет методология проектирования, которую можно определить, как набор методов, реализующих некоторые концепции, принципы проектирования и поддерживаемые какими-то средствами.

Проведём анализ методов проектирования информационных систем (Таблица 2.3.1).

Таблица 2.3.1 Классификация методов проектирования ИС

	Каноническое проектирование	Индустриальное автоматизированное проектирование	Индустриальное типовое проектирование
По степени автоматизации	ручные	компьютерные	компьютерные
По степени использования типовых проектных решений	оригинальные (индивидуальное)	оригинальные (индивидуальное)	типовые сборочные
По степени адаптивности проектных решений	методы реконструкции	методы параметризации	методы реструктуризации

По степени автоматизации методы проектирования делятся на ручные и компьютерные.

Ручные методы — это методы, при которых проектирование осуществляется без использования специальных программных средств, а программирование осуществляется на алгоритмических языках.

Компьютерные методы подразумевают использование специальных программных средств, в результате чего генерируется программный код или предлагается типовое проектное решение.

По степени использования типовых проектных решений различают следующие методы:

1. Оригинальные или индивидуальные методы, при которых проектные решения разрабатываются с самого начала, используя индивидуальный подход к решению поставленных задач;
2. Типовые методы предполагают применение готовых типовых решений или программных модулей.

По степени адаптивности проектных решений выделяют три метода:

1. Метод реконструкции, при котором происходит перепрограммирование программных модулей для адаптации проектных решений;
2. Метод параметризации заключается в настраивании проектных решений в соответствии с изменяемыми параметрами;
3. Метод реструктуризации модели заключается в изменении модели проблемной области, в результате чего, с использованием специализированных программных продуктов, происходит генерация проектных решений заново.

Из проведенного анализа видно, что при выборе определенной технологии применяются различные методы проектирования. Однако каноническое проектирование является отправной точкой для всех последующих технологий, т.к. их применение не исключает использование канонического проектирования. Следовательно, для разработки геоинформационной системы управления территориями заповяря на основе гетерогенных баз данных оптимальным решением будет выбор индустриального автоматизированного проектирования с применением методов канонического проектирования.

Сложность построения подобной системы состоит в применении и смешении различных технологий создания предметных информационных систем.

В геоинформационной системе управления территориями на основе гетерогенных баз данных необходимо сочетать геоинформационные технологии и технологии создания систем управления, что неразрывно влечёт за собой создания методики проектирования подобных систем. Достаточно вопросов создаёт и гетерогенность баз данных.

Изначально в проект должны быть заложены технологии, обеспечивающие процесс функционирования системы:

1. Нарращивание, изменение и распределение автоматизированных функций системы, конфигурирование системы;
2. Возможность автоматического обмена информацией между объектами как внутри системы, так и с внешними объектами;
3. Возможность динамической перестройки системы;
4. Автоматическую модернизацию системы, её компонентов при утрате актуальности информации в базах данных;

5. Взаимозаменяемость сервисов, для бесперебойной работы системы;
6. Защита системы от несанкционированного доступа, изменения и подмены данных;
7. Обеспечение взаимосвязи разноформатных, гетерогенных баз данных.

Применение подобных технологий требует дополнительного формального описания предметной области:

1. Описание задач автоматизации и их информационного обеспечения;
2. Описание структуры и объема информации, которая участвует в информационном обмене задач, объектов;
3. Описание требуемой структуры и объема информации при создании «персональных» баз данных пользователя;
4. Описание полномочий пользователя при работе с функциями и информацией;

Возможным решением для организации разработки подобной системы может быть ввод в состав информационного обеспечения объектов предметных областей и их структуры, входящих в систему гетерогенных баз данных, представленное в виде множеств иерархически упорядоченных реляционных таблиц – информационных ресурсов. Что предусматривает создание информационного хранилища с описанием атрибутов и их принадлежности к каждой гетерогенной базе данных. Описание помогает быстрее отправлять запрос к содержащей необходимую информацию базе данных. Тем самым задается информация, необходимая для построения информационной системы данного вида. При этом реализуются задание различных метасвойств информации в базе данных.

При введении понятия информационного ресурса необходимо обосновать состав, структуру, свойства и связи информационных ресурсов, которые адекватно описывают предметную область каждой отдельно взятой системы и из которой состоит геоинформационная система управления арктическими территориями.

Следующими действиями при проектировании подобных систем должно являться:

1. Описание распределения ресурсов и по местам сбора данных;
2. Описание обмена информацией между объектами автоматизации и с внешними системами в виде ресурсов;
3. Описание доступа пользователей к ресурсам, что позволяет продемонстрировать характер взаимодействия пользователя с системой;
4. Создание под хранение информационных ресурсов реляционной схемы базы данных.

Данные виды описаний можно рассматривать в контексте задач инфологического моделирования предметной области разрабатываемой системы. Для проведения такого моделирования существуют различные математические модели, методы и методики, в том числе и аппарат моделей сущность-связь или аппарат объектно-ориентированного анализа. В связи с этим разработаны и повсеместно применяются CASE-средства, которые могут обеспечить работу с данными математическими аппаратами. Так топология связей сущность-связь позволяет вычислить группировки сущностей, связанные между собой, которые рассматриваются в контексте единого целого как объекты предметной области. Такую информацию можно получить из анализа родительских и дочерних связей.

Немаловажным остается тот факт, что за основу геоинформационной системы управления территориями заповяря взята именно функциональность и наглядность геоинформационных систем. Следовательно, основное место в методике проектирования данных систем лежат технологии создания геоинформационных систем, осложнённые распределенностью и гетерогенностью баз данных.

Требования, которые можно предъявить к созданию подобной геоинформационной системы заключаются в разделении системы на «ядро» и «надстройку», при этом «ядро» должно быть универсально для выполнения любых функций ГИС.

Любое проектирование начинается с технических и организационных вопросов, которые перерастают в системное проектирование, параллельно с которым выполняется проектирование программного обеспечения (Рисунок 2.3.1)



Рисунок 2.3.1 Универсальные этапы проектирования геоинформационных систем

Техническое проектирование, при реализации геоинформационной системы управления территориями Арктической территорией на основе гетерогенных баз данных должно содержать в себе следующие функции:

1. Отладка работы системы по предъявляемым требованиям;
2. Законченность оформления требований к содержанию и функционированию системы по отношению к пользователю;
3. Утверждение форматов данных, с которыми будет работать система, из чего вытекает анализ входящих в систему гетерогенных баз данных;
4. Формирование требований к системе, таких как гибкость и надежность;
5. Выбор программного обеспечения осуществления проектируемой системы;
6. Структуризация потоков информации в информационной системе.

После определения функций в техническом проектировании производится моделирование. Строятся следующие модели:

1. Модель потребностей данных, где отображается взаимосвязь наличия данных и возможностей проведения анализа;
2. Модель потребностей приложений, где система рассматривается как набор функций, которые она может выполнить.

При организационном проектировании разработчик должен предложить следующие решения для возникающих перед ним задач:

1. Предложения по превращению проектируемой системы в рентабельную;
2. Расчёт стоимости получаемой с помощью системы информации;
3. Предложения по упрощению адаптации геоинформационной системы;
4. Предложения как удешевить и упростить поддержку ГИС.

Особенности проектирования подобной системы заключаются в формировании сложного интерфейса работы с данными, в неизвестности окончательного набора функций разрабатываемой системы до окончания разработки, в обеспечении расширяемости системы, в обеспечении поддержки модульности и скриптового языка программирования, в больших размерах проектируемой системы.

Для уменьшения объема данных необходимых для проведения единовременного анализа можно произвести фрагментацию данных по следующим параметрам:

1. Фрагментация по координатам;
2. Фрагментация по наиболее часто используемым объектам.

Также можно предложить ограничение доступа к определенным зонам карт.

Проектирование геоинформационной системы управления территориями Арктики РФ должно состоять как минимум из четырёх основных шагов (Рисунок 2.3.2)



Рисунок 2.3.2 Шаги проектирования геоинформационной системы управления на основе гетерогенных баз данных

Начальное представление заключается в описании разрабатываемой системы, анализе осуществимости системы, возможно ли вообще создать такую геоинформационную систему с требуемыми параметрами. Сюда входят описание не только предметной области системы, но и описание предметных областей включенных в неё гетерогенных баз данных, а также описанные выше информационные ресурсы системы, организованные в виде реляционной базы данных.

Концептуальное представление основано на определении требований и проектировании виртуальной интегрированной базы данных. Распределенность и гетерогенность баз данных включающихся в геоинформационную систему предусматривает место объединения и приведения к общему формату получаемой информации. Виртуальная интегрированная база данных позволяет создать на основе разноформатных локальных баз данных обобщенную, приведенную к единому формату виртуальную БД, которая может быть представлена в виде реляционной для упрощённого подключения ГИС-приложения.

Детальное представление системы заключается в отображении разрабатываемого проекта на конкретный ГИС-пакет, если используются готовые программные продукты геоинформационных систем. Или создание собственного приложения для отображения представленной информации в виртуальной базе данных и реализации необходимых функций.

Реализация системы предусматривает подключение базы данных к ГИС-приложению и программирование всех предусмотренных для выполнения функций системы.

Если рассматривать модель проектирования для геоинформационной системы управления на основе гетерогенных баз данных, то лучше остановиться на спиральной (Рисунок 2.3.3). Именно она предусматривает углубление и последовательную конкретизацию деталей проекта и в результате выбирается обоснованный вариант, который удовлетворяет действительным требованиям заказчика и доводится до реализации.

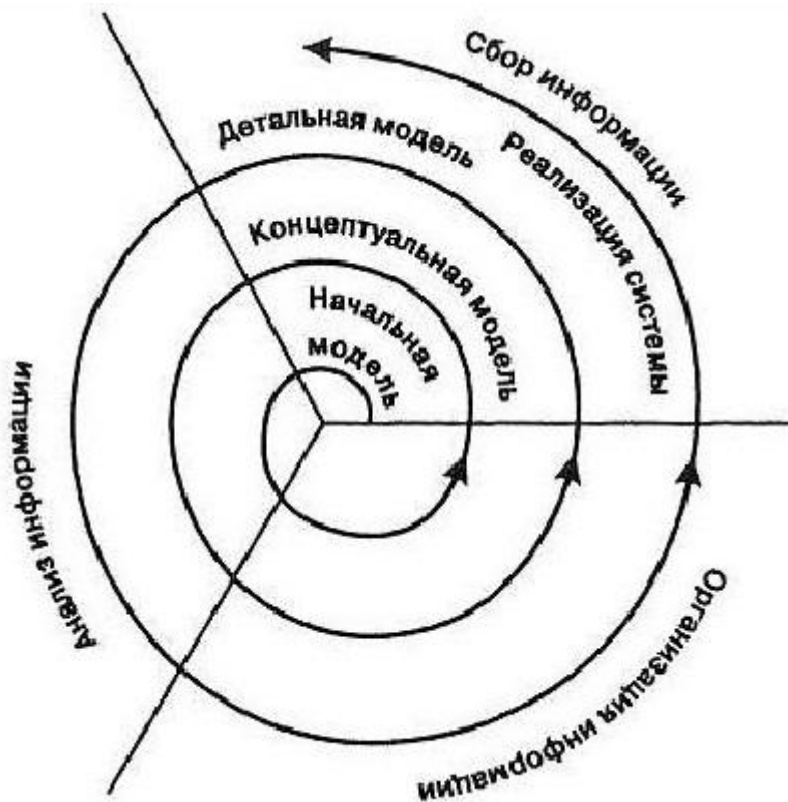


Рисунок 2.3.3 Спиральная модель проектирования геоинформационной системы управления на основе гетерогенных баз данных

В данной модели также прослеживаются этапы проектирования геоинформационной системы.

При создании крупномасштабной и дорогостоящей геоинформационной системы необходимо учитывать точки остановок в процессе проектирования ГИС (Рисунок 2.3.4).

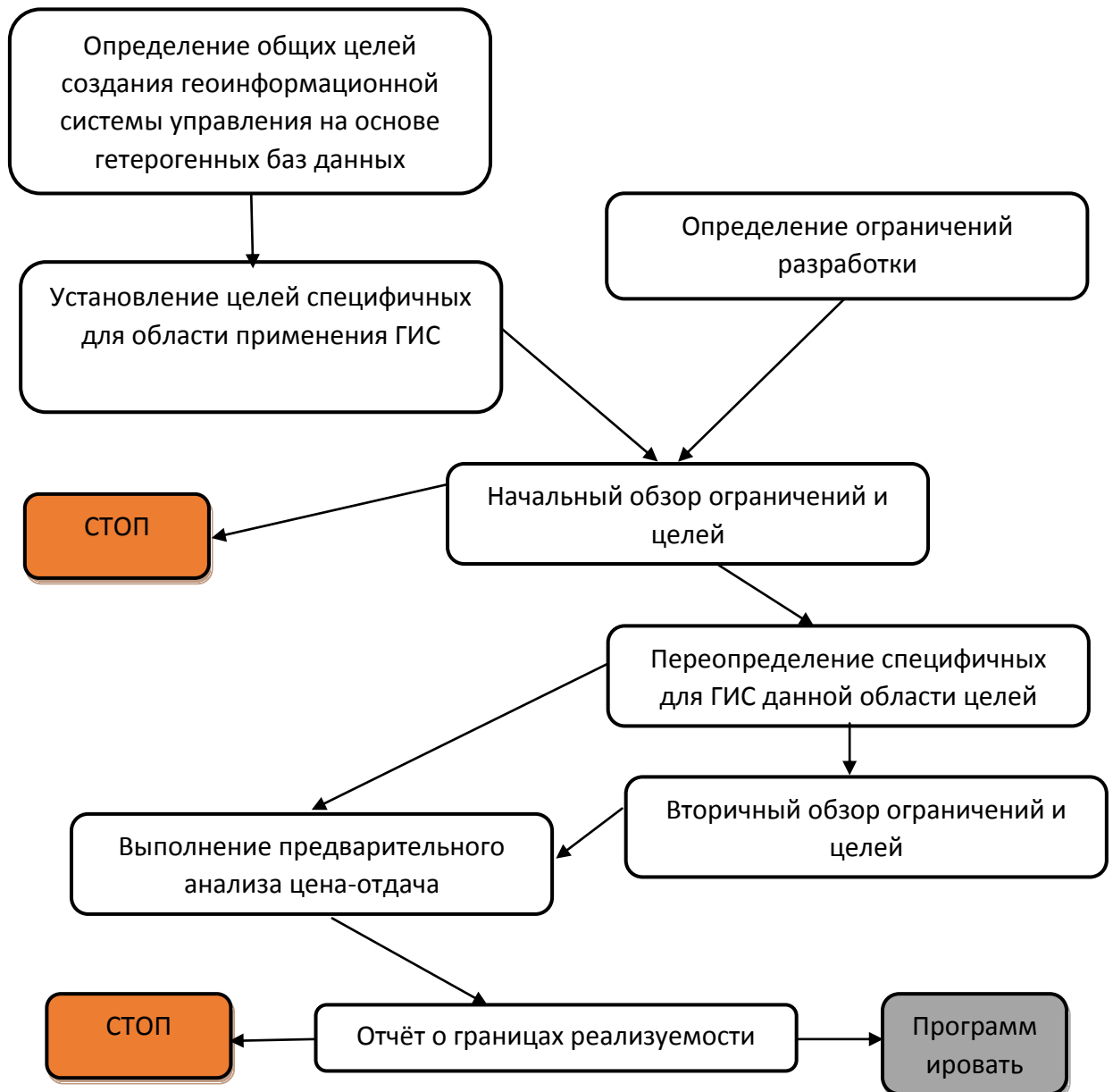


Рисунок 2.3.4 Точки остановок в процессе проектирования ГИС на основе гетерогенных данных

Применение подобной схемы точек остановки в процессе проектирования геоинформационной системы помогает максимально эффективно и безошибочно разрабатывать структуру и наполнение необходимой информационной системы.

Вывод по главе

В данной главе были даны определения информационной системы при системном подходе, структурной модели системы. Структурная модель информационной системы

наглядно демонстрирует, каким образом движутся внутри системы информационные потоки, как устроена физическая и логическая составляющая системы.

Рассмотрен математический аппарат расчета нагонных наводнений в Санкт-Петербурге, использованный при компьютерного моделирования процесса и сделан вывод, что данный метод прогнозирования может быть применим для моделирования процесса прогнозирования скорости ветра в Арктическом регионе.

Также были рассмотрены современные проблемы создания геоинформационной системы, осложненной гетерогенностью и распределенностью баз данных и представлена разработанная модель геоинформационной системы с препроцессорной подготовкой гетерогенных данных, охватывающая все слои представления информации: данные, обработка, интерфейс. Разработаны принципы и средства, способные эффективно размещать и синхронизировать доступ к пространственным геоданным в различных форматах без их явной конвертации. Для объединения и преобразования данных к желаемому виду пользователя используется «виртуальный процессор данных», в рамках которого «блок подготовки данных» позволяет получить доступ к любым источникам информации, необходимым для принятия управленческого решения.

В данной главе представлена модифицированная методика проектирования информационно-управляющей системы на основе распределенных гетерогенных баз данных, необходимая при реализации подобной системы. Было принято решение описания информационных ресурсов системы в виде реляционной базы данных. Поэтапность проектирования позволяет избежать ошибок и предусмотреть ситуации прекращения функционирования подобной системы.

Модифицированная методика проектирования позволяет реализовать методы и подходы создания геоинформационной системы управления на основе гетерогенных баз данных.

3. Информационная система обработки геоданных в Арктическом региона РФ

Исходя из некоторых оценок, 80% всех существующих данных относятся к географическим или пространственным данным. Можно упомянуть огромное количество информации получаемой при фотографировании земной поверхности со спутника. Эти данные используются большим количеством разнообразных групп пользователей в исследованиях, принятии решений, в быту по средствам применения геоинформационных систем.

Геоинформационные системы используются как промежуточное средство между данными привязанными к географической карте и исследователем. Применение ГИС в управление неразрывно связано с обработкой геоданных и выработки соответствующих рекомендаций в зависимости от ситуации.

3.1. Математическая модель оценки исследуемых геоданных

Геоинформационная система управления Арктической территорией РФ на основе гетерогенных баз данных предусматривает использование различных разноформатных локальных баз данных, территориально расположенных на расстоянии друг от друга.

Для принятия решений необходима своевременная и актуальная информация по искомому вопросу. В данном случае предметом исследования является приполярная территория, находящаяся в ведении России, где непосредственно расположен судоходный путь между Европейской частью России и Дальним Востоком.

Для ведения хозяйственной деятельности в Арктике необходимо учитывать климатические условия региона, выраженные в определенных критериях.

Для создания пробной версии ГИС управления на основе гетерогенных баз данных было выбрано 2 параметра: плотность льда и скорость ветра, для которых были применены методы математического моделирования. В условиях арктического климата ветер 10 м/с при температуре минус 5° соответствует воздействию на организм двадцатиградусного мороза, к тому же от скорости ветра зависят такие явления как бури, шторма и метели. Плотность льда в свою очередь – это главный показатель препятствия для прохода судов и уровня безопасности выхода на лед при рыболовстве. Для выбора оптимальных маршрутов плавания судов необходима оперативная ледовая информация, охватывающая изменения климата и площадь распространения льда в Арктике.

Все данные хранятся и поступают в систему из различных источников, из различных баз данных. Наибольшую информацию, включаемую в разрабатываемую систему, будут нести спутниковые снимки и данные представляемые ими. Тем самым источником информации для предполагаемой системы будут являться массивы данных со спутников Земли, представленные в виде файла с расширением .xps.

Одним из составляющих модулей геоинформационной системы является математическая обработки гетерогенных данных в рамках верификации геоинформационной системы, которая позволяет одновременно использовать данные для оценок вероятностных характеристик двух параметров (силы ветра и плотности льда), выраженная в схеме взаимосвязи данных (рисунок 3.1.1).

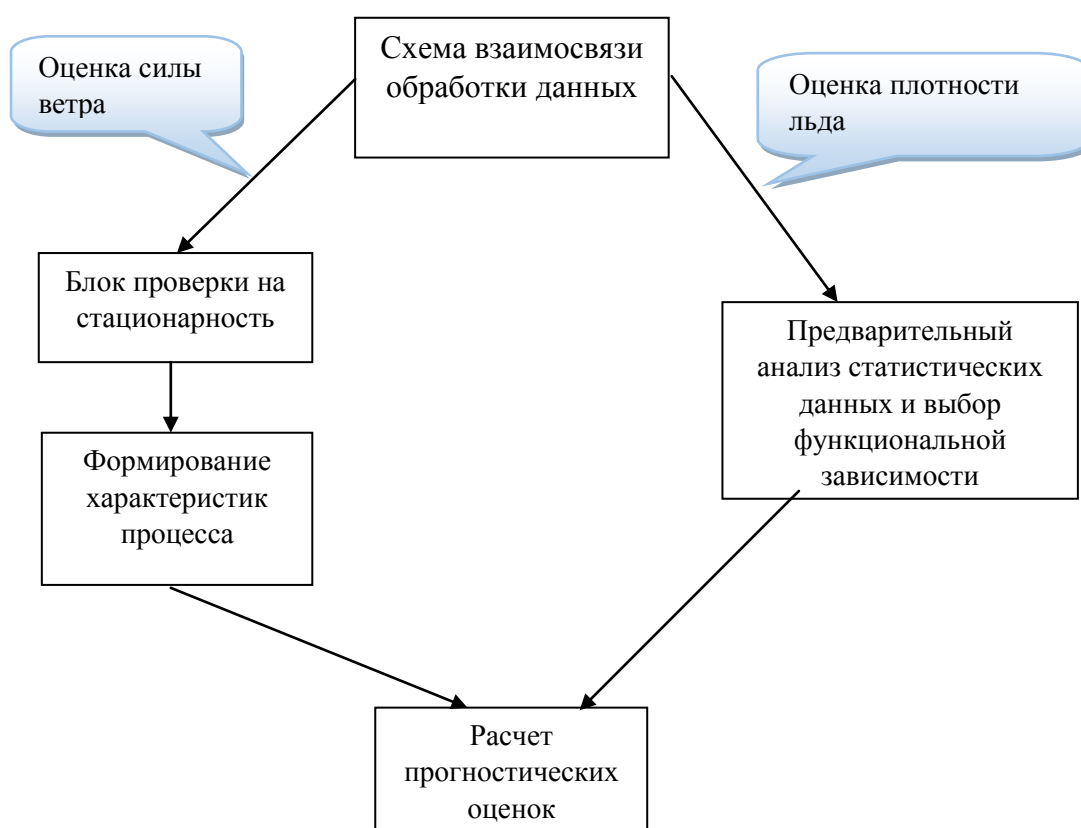


Рисунок 3.1.1 Схема взаимосвязи данных в математической модели

Геоинформационное прогнозирование представляет собой набор методов получения и анализа информации об окружающей среде и разработки прогнозных оценок для поддержки принятия решений [63]. Целью геоинформационного прогнозирования является снижение уровня неопределенности при принятии решений. В данном случае принимается решение о возможности прохождения судов по заданному пути, а также осуществлении других видов хозяйственной деятельности в Арктике.

При решении задач геоинформационного прогнозирования вся используемая информация подразделяется на два класса: первичная и вторичная. Первичные данные

получают в результате сбора информации об объектах местности различными технологиями: полевые методы, картометрические методы, дистанционные методы и т.д. На основе этих данных, после их унификации, создают наборы первичных цифровых моделей – местности, объектов, явлений, то есть разрабатываются базы данных, описывающие выявленные явления. Что и было сделано в результате обработки информации о плотности льда на приполярной территории, предоставленной лабораторией спутниковой океанографии РГГМУ [61]

Вторичные данные – данные, полученные в результате обработки первичных данных или собранные ранее из внутренних или внешних источников для других целей. Это информация, хранимая в архивах, базах данных, других ГИС, а также распространяемая по сети Интернет. Кроме того, под вторичными данными понимают модели, построенные по результатам обобщения и преобразования первичных моделей. К таким моделям относятся динамические ряды, результаты классификации или кластерного анализа и т.д.

Вторичные данные помогают выявить скрытые факторы, взаимосвязи, которые не были обнаружены в первичных данных.

После сбора данных необходимо их преобразовать к виду, отвечающему двум требованиям: удобному для обработки в геоинформационной системе; удобному для анализа при решении поставленной задачи прогнозирования.

Выделяют следующие основные функции преобразования данных: обобщение, определение концепции, коммуникация (перевод результатов статистического анализа на понятный для пользователя язык), экстраполяция (определение степени соответствия полученных результатов всей совокупности) [64].

Результатом геоинформационного прогнозирования являются не только статистические данные (обычный прогноз), но и тематические карты, цифровые модели динамики явлений, трехмерные модели объектов, векторные изображения полей или динамики явлений. Все эти данные получают на основе различных логико-математических методов прогнозирования.

Можно выделить несколько групп математических методов, используемых при проведении геоинформационного прогнозирования:

1. Статистические методы обработки информации.
2. Многомерные методы (в первую очередь факторный и кластерный анализы). Они используются для обоснования решений, в основе которых лежат многочисленные взаимосвязанные переменные.
3. Регрессионные и корреляционные методы. Они используются для установления взаимосвязей между группами переменных, описывающих маркетинговую деятельность.

4. Имитационные методы. Они применяются тогда, когда переменные, влияющие на реальную ситуацию (например, описывающие динамику), не поддаются определению с помощью аналитических методов.

5. Методы статистической теории принятия решений. Используются для стохастического описания ситуации.

6. Детерминированные методы исследования операций (в первую очередь линейное и нелинейное программирование). Эти методы применяют тогда, когда надо найти оптимальное решение.

Геоинформационные данные носят временной характер. При необходимости прогноза на определенный момент времени, в качестве исходных данных обычно используют временные ряды или временные модели.

При проведении периодически повторяемых наблюдений образуются наборы данных, характеризующиеся сбором информации об объектах земной поверхности через определенные периоды времени. В этом случае говорят о накоплении данных в виде временных рядов. Временные ряды представляют собой упорядоченные во времени наборы изменений каких-либо характеристик исследуемого объекта, процесса.

Временными рядами называют последовательности изменений $x(t1), x(t2), \dots, x(tn)$, упорядоченных в фиксированные моменты времени $t1, t2, \dots, tn$ [65].

В отличие от анализа случайных выборок анализ временных рядов основывается на предположении, что последовательные значения в наборе данных получены, как правило, через равные промежутки времени – периоды, которые могут быть различными: суточными, недельными, месячными, годовыми и т.д. По существу, временной ряд описывает непрерывное явление, протекающее во времени. Но информация об этом явлении поступает выборочно, вследствие чего реальный непрерывный процесс описывается в виде дискретных наборов данных.

Существуют три основные цели анализа временных рядов:

- построение формальной модели, позволяющей восстанавливать события в виде непрерывного процесса (переход от дискретной модели к непрерывной);
- установление механизма, определяющего динамику ряда (описание реального явления с помощью реально объясняющих и результативных переменных);
- прогнозирование явления или его анализ в ретроспективе (получение значений временного ряда по настоящим и прошлым значениям).

Эти цели требуют построения модели ряда и ее идентификации. Построение модели ряда означает, что ряд формально описан. При этом возможно формальное построение модели $\xi(t)$,

основанное на подборе математических зависимостей между параметрами ряда, безотносительно к механизму их формирования $\xi(t) = \xi(x(t1), x(t2), \dots, x(tn))$.

Эта модель называется формальной, цель ее построения – связать наилучшим образом полученные дискретные данные.

Возможно неформальное построение модели, основанное на описании механизма z протекания реального явления с введением параметров, описывающих реальные характеристики ai и bj , влияющие на изменение процесса или явления

$$F(ai, t) = \xi(x(t1), x(t2), \dots, x(tn), ai, bj), i = 1, \dots, n; j = 1, \dots, m.$$

Идентификация означает выбор в модели наблюдаемых bj и вычисляемых ai параметров, определение наблюдаемых параметров на основе экспериментальных данных. Как только модель ряда определена, с ее помощью можно интерпретировать рассматриваемые данные. Она позволяет, не вникая в механизм формализации, экстраполировать в перспективу или в ретроспективу и тем самым проводить анализ будущих или прошедших параметров явления. [42]

При оценке вероятности наступления какого-либо события можно воспользоваться известными методами оценки вероятностных характеристик процессов. Один из таких методов был использован при моделировании оценки вероятности возникновения нагонных наводнений в городе Санкт-Петербург, описанный в параграфе 2.1.2, который представлен временным рядом.

Возможность применения данного метода для оценки вероятности превышения силы ветра заданного порога на период времени Δt , зависит от удовлетворения следующим условиям:

1. Величина значения исследуемого параметра климатического явления $x(t)$ должна быть описана в виде случайных функций $x(t) = F(t, \theta)$, где t -время, θ – вектор случайных параметров, влияющих на изменение $x(t)$;
2. Функция распределения случайного процесса должна быть описана с достаточной достоверностью под нормальный закон распределения;
3. Существуют такие временные интервалы, на которых $x(t)$ является стационарным случайным процессом.

В большинстве случаев геоинформационные данные могут быть представлены в виде временных рядов или иных случайных процессов. В том числе изменение силы ветра, также является функцией изменения времени и ряда параметров, которые носят как случайный, так и детерминированный характер.

Накопленный объем статистических данных по рассматриваемому показателю настолько велик, что можно утверждать о возможности описания с помощью нормального закона

распределения. Выносятся предположения что существуют такие интервалы времени на которых исследуемый показатель является стационарным случайным процессом.

Всё выше написанное даёт возможность применения известной модели для решения задачи оценки вероятности превышения силы ветра заданного уровня. В данном случае было бы логично критериальные оценки проводить с использованием шкалы Бофорта (Таблица 3.1.1), в которой существуют интервал скорости, при котором присваивается тот или иной бал.

Таблица 3.1.1 Классификация силы ветра. Шкала Бофорта

Баллы	Наименование	Скорость ветра	Описание
0	штиль	0-0,2км/час	Зеркально гладкое море, практически неподвижное. Волны практически не набегают на берег. Вода больше похожа на тихую заводь озера нежели на морское побережье. Над поверхностью воды может наблюдаться дымка. Край моря сливается с небом так, что границы не видно.
1	тихий	0,2-1,5 м/с	На море легкая рябь. Высота волн достигает до 0,1 метра. Море по-прежнему может сливаться с небом. Чувствуется легкий, почти незаметный ветерок.
2	легкий	1,6-3,3 м/с	Небольшие волны, высотой не более 0,3 метра. При таком ветре флюгер начинает двигаться.
3	слабый	3,4-5,4 м/с	Легкое волнение на воде, изредка появляются барашки. Средняя высота волн до 0,6 метров. Хорошо заметен слабый прибой.
4	умеренный	5,5 - 7,9 м/с	Флюгер крутится непрерывно, гнутся тонкие ветви деревьев. Море беспокойное, во многих местах видны барашки. Высота волн до 1,5 метра.
5	свежий	8 - 10,7 м/с	Почти все море покрыто белыми барашками. Высота волны 2 метра. Качаются ветки и тонкие стволы деревьев.
6	сильный	10,8 - 13,8 м/с	Море во многих местах покрыто белыми гребнями. Высота волн достигает 4х метров, средняя высота 3 метра. Гнуться тонкие стволы деревьев, и толстые сучья деревьев, гудят телефонные провода
7	крепкий	13,9 - 17,1 м/с	Море покрыто белыми пенистыми гребнями, которые время от времени срываются ветром с поверхности воды. Высота волн достигает 5,5 метров, средняя высота 4,7 метров. Качаются средние стволы деревьев, гнутся сучья.
8	очень крепкий	17,2 - 20 м/с	Сильные волны, на каждом гребне пена. Высота волн достигает 7,5 метров, средняя высота 5,5 метров. Идти против ветра трудно. Ломаются тонкие сучья деревьев.
9	шторм	20,8 - 24,4 м/с	Высокие волны на море, достигающие 10 метров; средняя высота 7 метров. Гнутся большие деревья, ломаются средние ветки. Ветер срывает плохо укрепленное покрытие с крыш.

Ба лл ы	Наименова ние	Скорость ветра	Описание
10	сильный шторм	24,5 - 28,4 м/с	Море белого цвета. Волны обрушаются на берег или о скалы с грохотом. Максимальная высота волн 12 метров, средняя высота 9 метров. Ветер срывает крыши, значительные повреждения строений.
11	жестокий шторм	28,5 - 32,6 м/с	Высокие волны достигают 16 метров, при средней высоте 11,5 метров. Сопровождается большими разрушениями на суше.
12	ураган	32,6 м/с и более	Серьезные повреждения капитальных строений. Высота волн более 16 метров.

В качестве случайного события при данной постановке задачи может быть рассмотрена задача превышения силы ветра одного из уровней описанного баллами. Чаще нас будет интересовать случай выхода за пределы 11 балла.

Построим случайную функцию, описывающую изменение силы ветра во времени $S(t)$ (Рисунок 3.1.2).

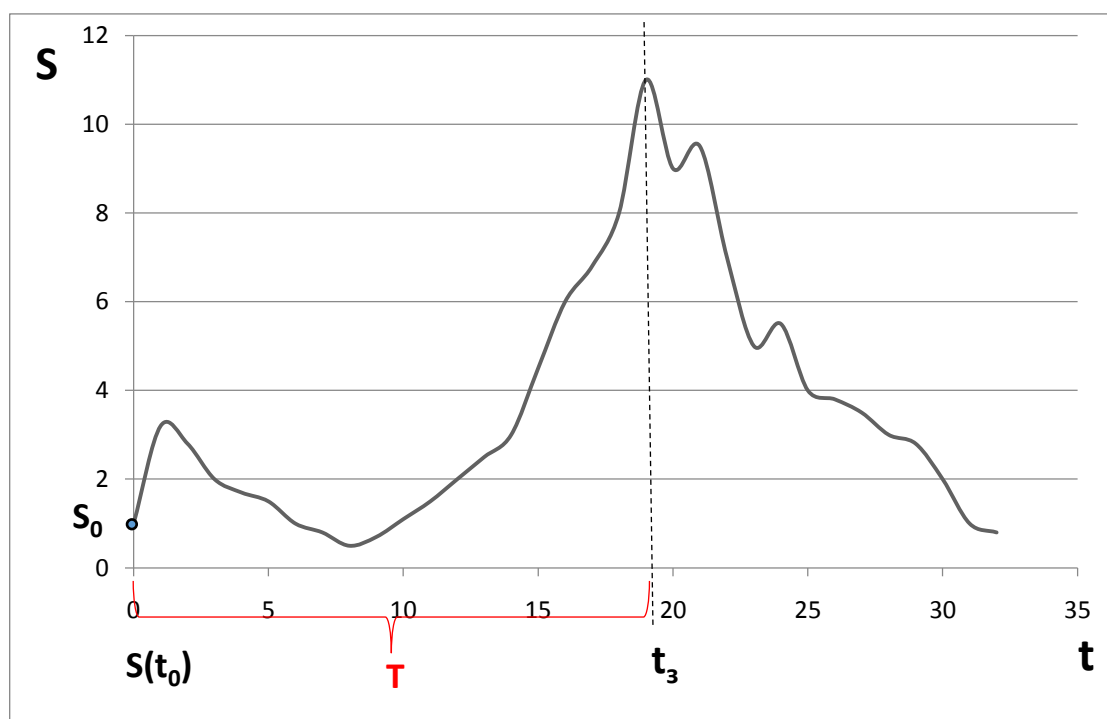


Рисунок 3.1.2 Случайная функция, описывающая изменение силы ветра во времени

При известном случайном процессе, можно определить вероятность невыхода силы ветра за промежуток, не превышающий выбранный балл по шкале Бофорта по следующей формуле:

$$P_i(t_3) = P(S(t_3)/S(t_0) = S_0 \leq \delta_i), \quad (3.1.1)$$

где:

$P_i(t_3)$ – вероятность невыхода силы ветра за пределы i -го балла по шкале Бофорта,

δ_i – границы шкалы Бофорта на i -ом срезе,

S_0 – начальное значение силы ветра,

$S(t_3)$ – случайная функция изменения силы ветра на заданный момент времени t_3

$S(t_0)$ – случайная функция изменения силы ветра на начальный момент времени t_0

Используя оценки, предложенные в работе профессора Истомина Е.П. [53] мы получим:

$$P_i(t_3) = \Phi\left(\frac{\delta_i - m_s - r_s(0, t_3) \cdot (s_0 - m_s)}{\sigma_s \sqrt{1 - r_s^2(0, t_3)}}\right) - \Phi\left(\frac{-m_s - r_s(0, t_3) \cdot (s_0 - m_s)}{\sigma_s \sqrt{1 - r_s^2(0, t_3)}}\right) \quad (3.1.2)$$

где: m_s – математическое ожидание случайного процесса;

σ_s – среднеквадратическое отклонение;

$r_s(0, t_3)$ – автокорреляционная функция.

Математическое ожидание времени невыхода случайного процесса за установленные границы в периоды измерения от t_0 до t_3 может быть оценено в соответствии с использованием различных методик. В данной работе использовалась методика, описанная в работе [53], где рассматривалась модель без учета взаимной корреляции между двумя точками.

Для исследования берется минимум 2 точки x и y , такие чтобы взаимная корреляция между ними была тождественно равна нулю:

$$r_{x,y} \equiv 0 \quad (3.1.3)$$

Если же берется больше точек, тогда взаимная парная корреляция тождественно равна нулю:

$$r_{x,y} = r_{x,z} = r_{z,y} \dots \equiv 0. \quad (3.1.4)$$

Поскольку координатные точки, по которым производится прогноз существенно отдалены друг от друга, то можно считать общую вероятность наступления прогнозируемого явления как произведение вероятностей появления этого события в каждой из точек.

Однако данный алгоритм, в связи с большими допущениями, требует проверки статистической гипотезы на принадлежность исходных данных выбранной модели. В данном случае осуществляется проверка с использованием эффективного способа проверки согласия между предсказаниями модели и опытными данными, предложенный Карлом Пирсоном в 1900 году.

Предложенный им «хи-квадрат критерий» – это самый важный и наиболее часто используемый статистический критерий. Большинство задач, связанных с оценкой неизвестных параметров модели и проверки согласия модели и опытных данных, можно решить с его помощью.

Параметры модели можно оценить разными способами оценки – «метод максимального правдоподобия», «метод моментов», «метод подстановки». Однако можно не привлекать никаких дополнительных средств и найти оценки параметров минимизируя расстояние Пирсона. В докомпьютерную эпоху такой подход использовался редко: при ручных расчетах он неудобен и, как правило, не поддается аналитическому решению. При расчетах на компьютере численная минимизация обычно легко осуществляется, а преимуществом такого способа является его универсальность. Итак, согласно «методу минимизации хи-квадрат», мы подбираем значения неизвестных параметров так, чтобы расстояние Пирсона стало наименьшим. (Кстати, изучая изменения этого расстояния при небольших смещениях относительно найденного минимума можно оценить меру точности оценки: построить доверительные интервалы.) После того как параметры и само это минимальное расстояние найдено опять требуется ответить на вопрос достаточно ли оно мало.

Общая последовательность действий такова:

1. Выбор модели (гипотезы H_0).
2. Выбор разрядов и определение вектора наблюдаемых частот O_i .
3. Оценка неизвестных параметров модели и построение для них доверительных интервалов (например, через поиск минимума расстояния Пирсона).
4. Вычисление ожидаемых частот E_i .
5. Сравнение найденной величины расстояния Пирсона χ^2 с критическим значением хи-квадрат $\chi^2_{\text{крит}}$ - наибольшим, которое еще рассматривается как правдоподобное, совместимое с H_0 . Величину, $\chi^2_{\text{крит}}$ мы находим из таблиц, решая уравнение $P(\chi^2_n > \chi^2_{\text{крит}}) = 1 - \alpha$,

где α – «уровень значимости» или «размер критерия» или «величина ошибки первого рода» (типичное значение $\alpha=0.05$).

Обычно число степеней свободы n вычисляют по формуле

$n = (\text{число разрядов}) - 1 - (\text{число оцениваемых параметров})$ Если $\chi^2 > \chi^2_{\text{крит}}$, то гипотеза H_0 отвергается, в противном случае принимается. В $\alpha \cdot 100\%$ случаев (то есть достаточно редко) такой способ проверки H_0 приведет к «ошибке первого рода»: гипотеза H_0 будет отвергнута ошибочно. [42]

Для прогнозирования вероятностных значений плотности льда было решено использовать алгоритмы, позволяющие решать задачи прогнозирования состояний при полной определенности исходных значений множество (методы математической статистики, метод наименьших квадратов, максимального правдоподобия и другие). Одним из таких методов являются оптимальные фильтры. Задачей фильтров является построение алгоритма, который осуществляет необходимое преобразование сигналов, поступающих на его вход. Наиболее универсальным фильтром для решение поставленной задачи является фильтр Калмана – Бьюси. Главная особенность фильтра в том, что он обновляет оценку текущего состояния системы при

поступлении новых данных, а затем предсказывает их будущее состояние. Сам же фильтр Калмана – Бьюси является рекуррентным алгоритмом, что дает преимущество реализации его на персональном компьютере.

Прогнозирование с помощью фильтра Калмана – Бьюси возможно только в случае, если прогнозируемые процессы в дискретном случае можно описать разностными уравнениями типа

$$x_t = \Phi(t, t-1)x_{t-1}, \quad (3.1.5)$$

где $\Phi(t, t-1)$ - переходная матрица, характеризующая структуру x_t , при этом наблюдения должны удовлетворять соотношению

$$z_t = P_t x_t + \varepsilon_t, \quad (3.1.6)$$

где P_t - матрица ограничений на наблюдения; ε_t - ошибка наблюдений.

Для того чтобы решить задачу прогнозирования $y(t)$ с помощью фильтра Калмана – Бьюси, необходимо провести зависимость

$$\tilde{y}_i(t) = \sum_{j=0}^m a_{ij} \varphi_j(t), \quad i = \overline{0, n}, \quad (3.1.7)$$

где a_{ij} - случайные величины; $\{\varphi_j(t)\}_{j=0}^m$ - непрерывные детерминированные функции времени в соответствие с выражением (3.1.5).

Требуемые преобразования можно осуществить на основе расширения пространства состояний, то есть путем введения новых координат в пространство параметров, определяющих техническое состояние. Компонентами вектора x_t будут служить коэффициенты a_{ij} (элементы матрицы A выражения (3.1.7)), отсюда x_t можно сформировать на строк матрицы A , где его размерность будет $l = nm$, где n и m – число строк и столбцов матрицы. При этом $\Phi(t, t-1)$ - единичная матрица размером $l \times l$, а матрица P_t размера $n \times l$ состоит из компонентов вектора $\{\varphi_j(t)\}_{j=0}^m$ - детерминированного базиса модели (3.1.7). Построение P_t можно реализовать последовательным сдвигом по строкам на $i(m+j)$ транспонированного вектора φ (i - номер строки матрицы P_t) [55].

Решением уравнений (3.1.5) и (3.1.6) будут известные соотношения, определяющие фильтр Калмана – Бьюси. После преобразования этих соотношений в соответствии с постановкой задачи прогнозирования получим:

$$x_{t+1} = x_t + W_t(z_t - P_t x_t) \quad (3.1.8)$$

$$W_t = K_t P_t^T (P_t K_t P_t^T + R_t)^{-1}$$

$$K_{t+1} = K_t - W_t P_t K_t \quad (3.1.9)$$

где X_{t+1} - прогнозируемая на момент времени $t+1$ по результатам имеющихся к этому времени наблюдений оценка x (условное математическое ожидание x); X_t - прогнозируемая на момент времени t оценка x ; K_t - ковариационная матрица компонентов вектора X_t ; K_{t+1} - ковариационная матрица компонентов вектора X_{t+1} ; W_t - матрица передачи фильтра; Z_t - результаты наблюдений $y(t)$ в момент t ; R_t - ковариационная матрица ошибок измерений ε_t .

Для прогнозирования, то есть для решения выше приведенных уравнений, служат априорные оценки математических ожиданий – коэффициенты a_{ij} , их ковариационная матрица, результаты контроля $y(t)$ и ковариационная матрица ошибок измерений. Следует отметить, что безотносительно к форме распределения входного сигнала при заданном первом и втором моментах фильтр Калмана – Бьюси обеспечивает получение наилучших в среднеквадратическом смысле линейных оценок.

Для решения поставленной задачи будем использовать модель:

$$y(t) = a_0 + a_1 t + a_2 t^2, \quad (3.1.10)$$

Так как выше сказано, что фильтр использует рекуррентную обработку информации, наделяя данный метод адаптивными свойствами, в свою очередь позволяющие применять более простые модели $y(t)$, в частности модель вида:

$$y(t) = A \cdot F(t) \quad (3.1.11)$$

не выше второго порядка ($m = 2$).

Для подтверждения адекватности выбранной модели необходимо воспроизвести её на прикладном уровне. Для этого используется пакет программ математического моделирования MATLAB.

Входными данными для фильтра Калмана – Бьюси служат априорные оценки математических ожиданий коэффициентов a_{ij} и их ковариационная матрица, полученные с помощью использования модели (6). В среде MATLAB выглядит следующим образом:

Формируется математическое ожидание коэффициентов в данном случае $a_0 \ a_1 \ a_2$ – $a_2 \ a_1 \ a_0$ (поскольку функция возвращает значения именно в таком порядке)

```
n = 1;  
a0 = (p1(n) + p2(n) + p3(n))/3;  
n = 2;  
a1 = (p1(n) + p2(n) + p3(n))/3;  
n = 3;  
a2 = (p1(n) + p2(n) + p3(n))/3;  
Mx = [a0 a1 a2];
```

(коэффициенты p_1, p_2, p_3 формированы с помощью метода наименьших квадратов)

После чего формируется матрица ковариации коэффициентов $a_0 - a_2$

```
AfK = [p1; p2; p3];  
Kt = cov(AfK);  
Kt1 = cov(AfK);
```

В дальнейшем параметры и априорные коэффициенты будут меняться. Связанно это с периодическими изменениями наблюдаемого объекта, в данном случае концентрации льда относительно времени измерения, что дает возможность сделать экстраполяцию.

Для правильной работы фильтра Калмана – Бьюси относительно выражения (3.1.8) и выше упомянутых условий, матрица ковариаций K_t имеет размерность 3×3 , соответственно для построения необходимо реализовать аппроксимацию зависимости параметра по времени трижды для получения оценок коэффициентов a_{ij} . Аппроксимировать зависимость параметра по времени возможно с использованием метода экспоненциального сглаживания или метода наименьших квадратов, последний был применен в решение поставленной задачи.

Били взяты три точки на водной глади Приполярной территории, по которым с помощью спутникового зондирования Земли были получены данные о состоянии плотности льда. Информация была предоставлена лабораторией спутниковой океанографии, расположенной в Российском гидрометеорологическом университете, в городе Санкт-Петербург. В результате, при вводе данных в реализованный фильтр Калмана – Бьюси на MATLAB, получилась матрица ковариаций 3×3 следующего вида:

```
Kt1 =  
1.0e-03 *  
0.0023 -0.0142 0.0208  
-0.0142 0.0903 -0.1347  
0.0208 -0.1347 0.2041
```

Все данные концентрации льда за период наблюдения используемые в постановке задачи импортируются в MATLAB как CSV файл с помощью команды:

```
importfile('135_85.csv','135_85');
```

Метод интегрирования CSV файлов в MATLAB (в данной программной среде хорошо реализована поддержка файлов с расширением типа «значения, разделённые запятыми») значительно упрощает ввод большого количества параметров.

Как наглядный результат реализации фильтра Калмана – Бьюси в среде MATLAB с использованием данных концентрации плотности льда, полученных со спутниковой ГИС, служит рисунок 3.1.3.

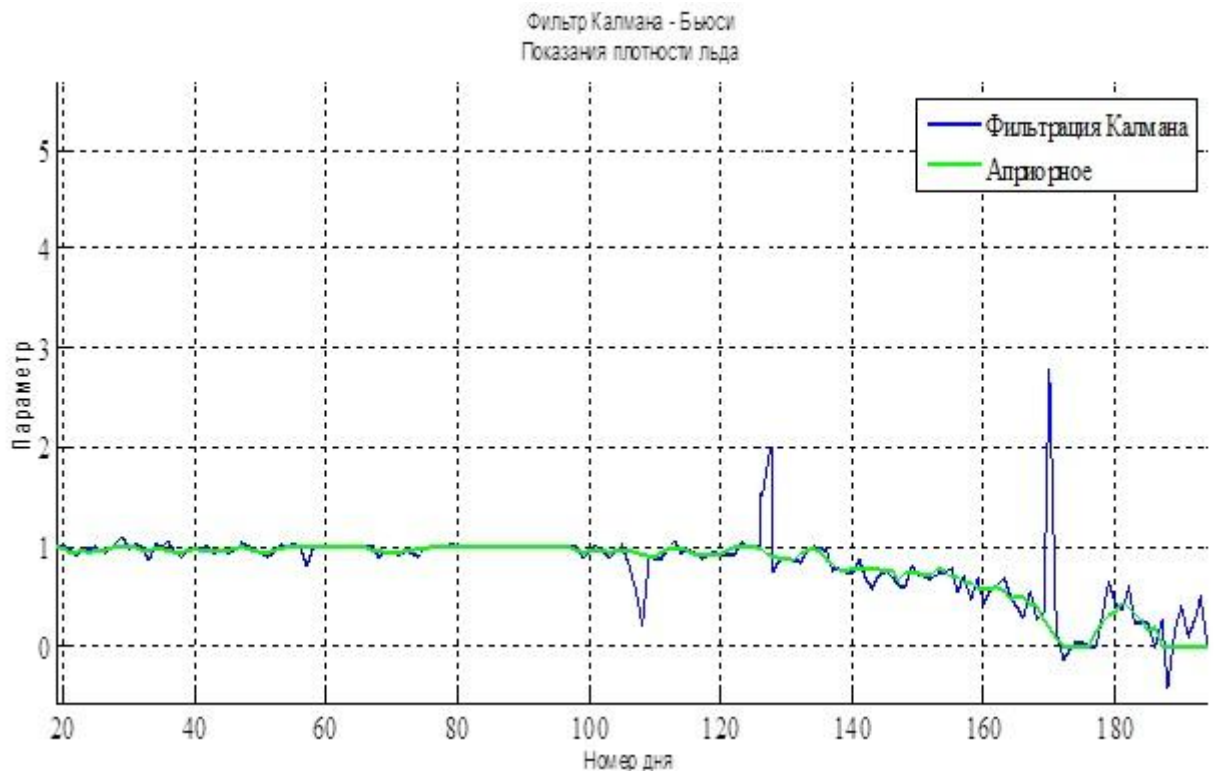


Рисунок 3.1.3. Прогноз изменения концентрации плотности льда на Приполярной территории с помощью фильтра Калмана – Бьюси по заданным трём точкам

Проанализировав график видно, что фильтр Калмана – Бьюси реализованный в среде MATLAB работает и является актуальным методом решения поставленной задачи, так как данные прогноза фильтра достаточно приближены к априорным значениям. Но, на графике очевидны образования достаточно серьезных всплесков. Эти отклонения прогнозирования фильтра от априорных значений вызваны тем, что данные о концентрации плотности льда, полученные со спутника имеют пропуски (нулевые значения) в определенные дни измерений. То есть при расчете нулевых значений получается очень маленькая погрешность. Матрица ковариаций вырождается, после чего не рассчитывается и на графике отображается в виде большого всплеска.

Для того чтобы решить данную проблему необходимо сделать проверку на ноль и в соответствии с этим увеличивать величину погрешности. Еще один способ в качестве устранения всплесков - применение восстановления пропусков наблюдений, основанный на

построении регрессионных уравнений с одним или несколькими аналогами, которые имеют как более продолжительный период наблюдения, так и данные в тот период времени, которые были пропущены при измерении с помощью спутника. Также для более точной реализации фильтра Калмана – Бьюси в среде MATLAB необходимо учитывать более детальные параметры управляющих воздействий, как пример - законы по которым меняется плотность льда.

Не смотря на выше сказанное, при ненулевых параметрах данная модель производит прогнозирование, за счет скопленных данных, с достаточно большой вероятностью, приближенным к исходным измерениям.

Исходя из проведенного исследования, можно сказать, что при достаточном и достоверном объеме данных, данный метод имеет положительный характер для дальнейшего более глубоко рассмотрения.

Полученные результаты уже сейчас могут оказать помощь в прокладке маршрутов для судов на Приполярной территории, как со стороны эффективности (до 30%), так и со стороны безопасности.

3.2. Моделирование геоинформационной системы «Оценка климатических параметров для ведения хозяйственной деятельности на арктической территории РФ»

Моделирование геоинформационной системы может осуществляться различными путями, как с использованием поэтапного моделирования в соответствии с ГОСТ, так и с помощью современных средств визуального моделирования (CASE – средств), существующих методологию объектно-ориентированного анализа и проектирования.

К CASE-технологиям относятся такие средства визуального моделирования как DFD, IDEF0, IDEF1, IDEF2 диаграммы и язык UML. Наиболее оптимальным для построения геоинформационной системы является применение языка UML, так как в результате проектирования программное средство на его основе способно сгенерировать код базы данных и приложения на предусмотренном программой языке программирования.

Язык UML представляет собой общецелевой язык визуального моделирования, который разработан для спецификации, визуализации, проектирования и документирования компонентов программного обеспечения, бизнес-процессов и других систем. Язык UML

является достаточно строгим и мощным средством моделирования, которое может быть эффективно использовано для построения концептуальных, логических и графических моделей сложных систем различного целевого назначения. Этот язык вобрал в себя наилучшие качества и опыт методов программной инженерии, которые с успехом использовались на протяжении последних лет при моделировании больших и сложных систем. [67]

При использовании методологии объектно-ориентированного анализа и проектирования моделью сложной системы можно считать некоторое количество взаимосвязанных представлений. Каждое такое представление объективно отражает все моменты поведения или структуры информационной системы. Из всех возможных представлений наиболее общими считают статические и динамические, которые могут подразделяться на более детальные представления.

Принцип иерархического построения моделей сложных систем предписывает рассматривать процесс построения моделей на разных уровнях абстрагирования или детализации в рамках фиксированных представлений.

При построении модели сложной системы в первую очередь строится самое общее представление, которое относится к начальному или концептуальному уровню проектирования. На этом уровне модель не включает в себя множество деталей и моментов моделируемой системы. Все дополнения осуществляются при создании представлений на логическом и физическом уровне.

Весь процесс объектно-ориентированного анализа и проектирования можно свести к последовательному созданию наиболее общих моделей и представлений, к примеру, на концептуальном уровне, до создания более частных и детальных представлений, на логическом и физическом уровнях. Так на каждом этапе проектирования модели дополняются всё большим количеством деталей и уточнений, что позволяет им более точно соответствовать предметной области реализуемой системы (Рисунок 3.2.1).



Рисунок 3.2.1 Общая схема взаимосвязей моделей объектно-ориентированного анализа и проектирования

Для концептуального представления или построения концептуальной модели системы в процессе ее проектирования и разработки используют диаграмму вариантов использования в языке визуального программирования UML. Создание диаграммы вариантов использования имеет следующие цели:

1. Определить общие границы и контекст моделируемой предметной области на начальных этапах проектирования системы;
2. Сформулировать общие требования к функциональному поведению проектируемой системы;
3. Разработать исходную концептуальную модель системы для ее последующей детализации в форме логических и физических моделей;
4. Подготовить исходную документацию для взаимодействия разработчиков системы с ее заказчиками и пользователями.

Начальным этапом моделирования геоинформационной системы «Оценки вероятности образования рисков для прохождения судов на Приполярной территории» является анализ предметной области, из чего вытекают требования к реализуемой системе и появляется возможность построения первой концептуальной модели в виде диаграммы вариантов использования (Рисунки 3.2.2 – 3.2.3).

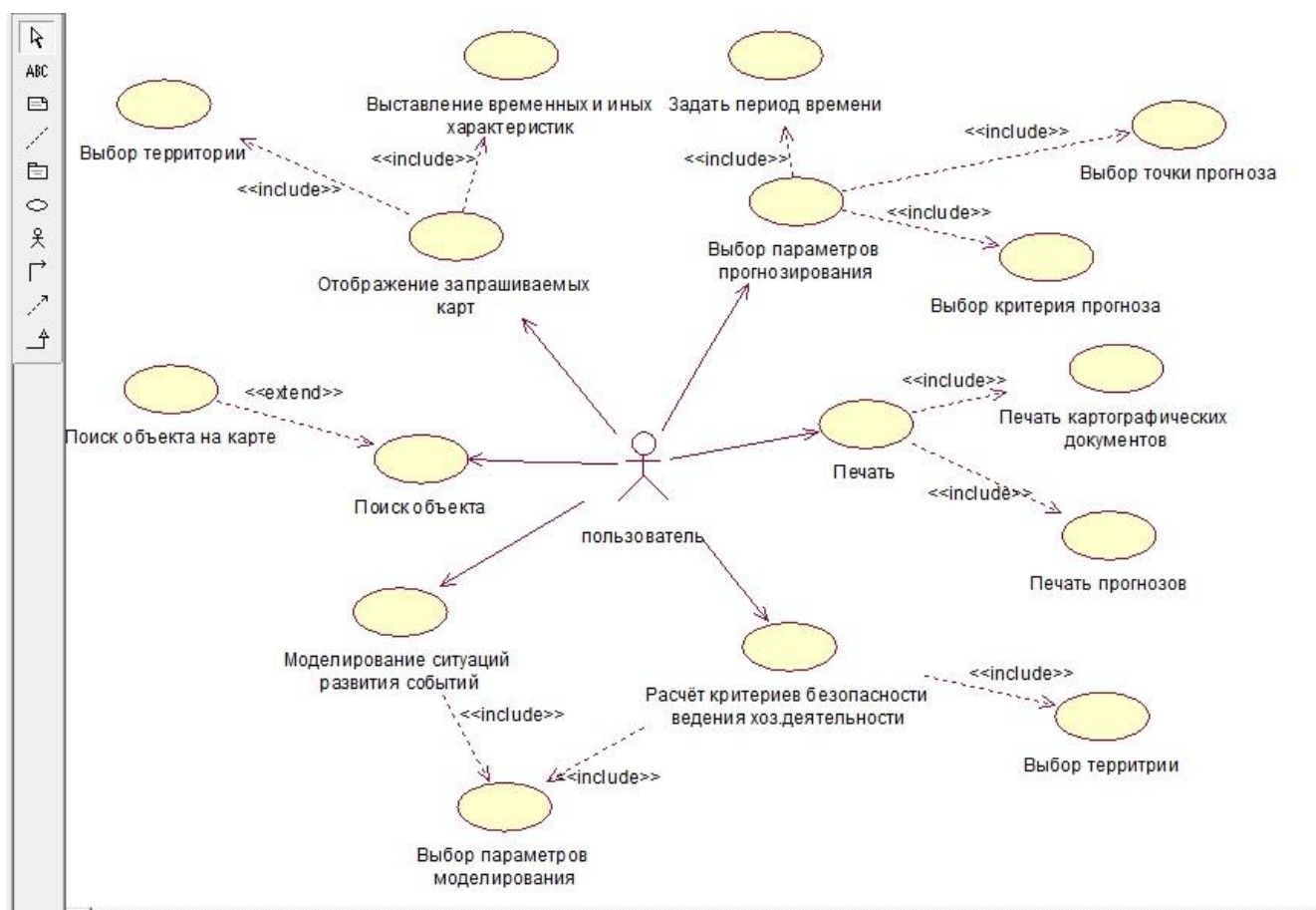


Рисунок 3.2.2 – Диаграмма вариантов использования для действующего лица – пользователь

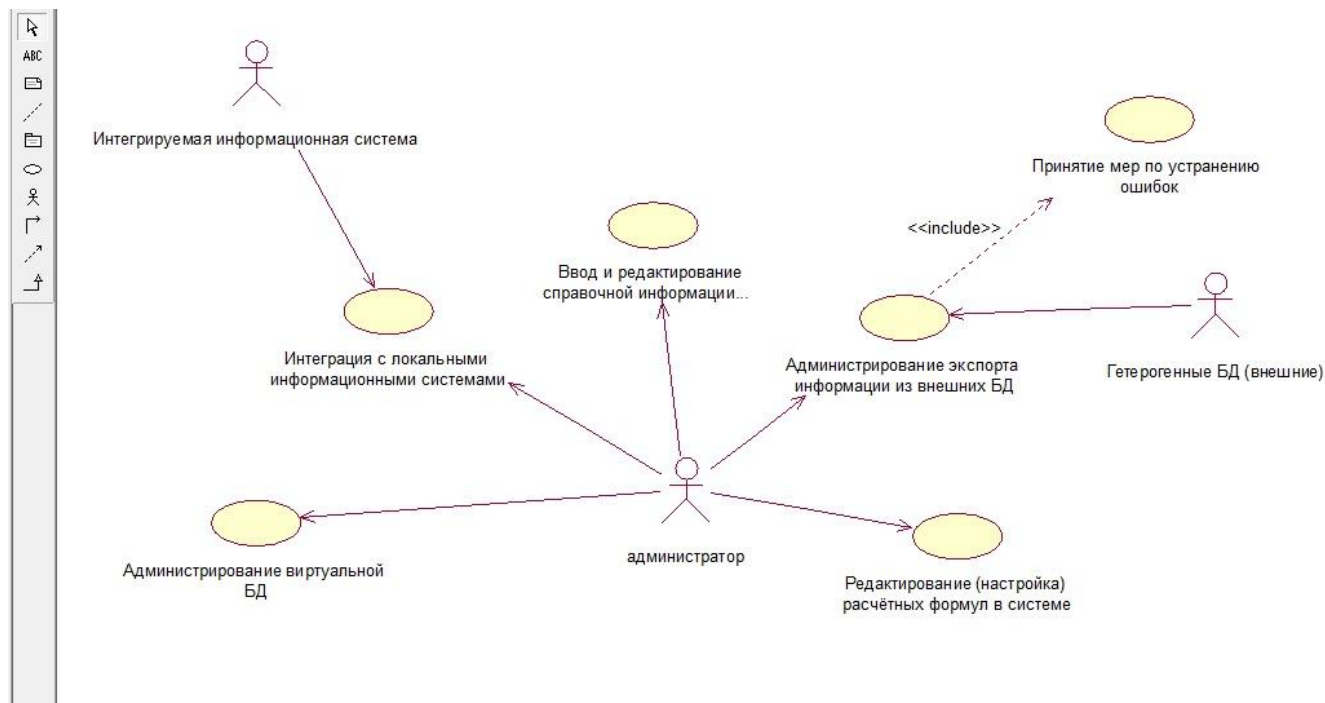


Рисунок 3.2.3 – Диаграмма вариантов использования для действующего лица – администратор

Действующими лицами, взаимодействующими непосредственно с геоинформационной системой, являются - пользователь и администратор, представленные в нотации языка UML в

виде актёров. Актёрами также выступают и интегрируемые информационные системы и гетерогенные базы данных, поскольку эти данные составляют, но не являются частью системы, а лишь влияют на неё.

Исходя из потребностей действующих лиц выделяются необходимые для их реализации варианты использования.

В данном случае действующие лица имеют различные потребности и соответственно имеют совершенно разный набор вариантов использования.

Следующим этапом в методологии объектно-ориентированного анализа и проектирования является разработка логической модели системы, которая выступает в виде диаграммы классов. Диаграмма классов отражает различные взаимосвязи между отдельными сущностями предметной области, такими как объекты и подсистемы, а также описывает их внутреннюю структуру и типы отношений. На данной диаграмме не указывается информация о временных моментах функционирования системы. Диаграмму классов можно считать развитием концептуальной модели проектируемой системы.

Диаграмма классов представляет статическую структуру модели проектируемой системы в терминологии классов объектно-ориентированного программирования. При упоминании о данной диаграмме, имеют в виду статическую структурную модель проектируемой системы, а именно графическое представление таких структурных взаимосвязей логической модели системы, которые не зависят от времени.

При моделировании геоинформационной системы «Оценки рисков для ведения хозяйственной деятельности на арктической территории РФ» была создана общая диаграмма классов, включающая все непосредственные компоненты и объекты необходимые для реализации предметной области (Рисунок 3.2.4).

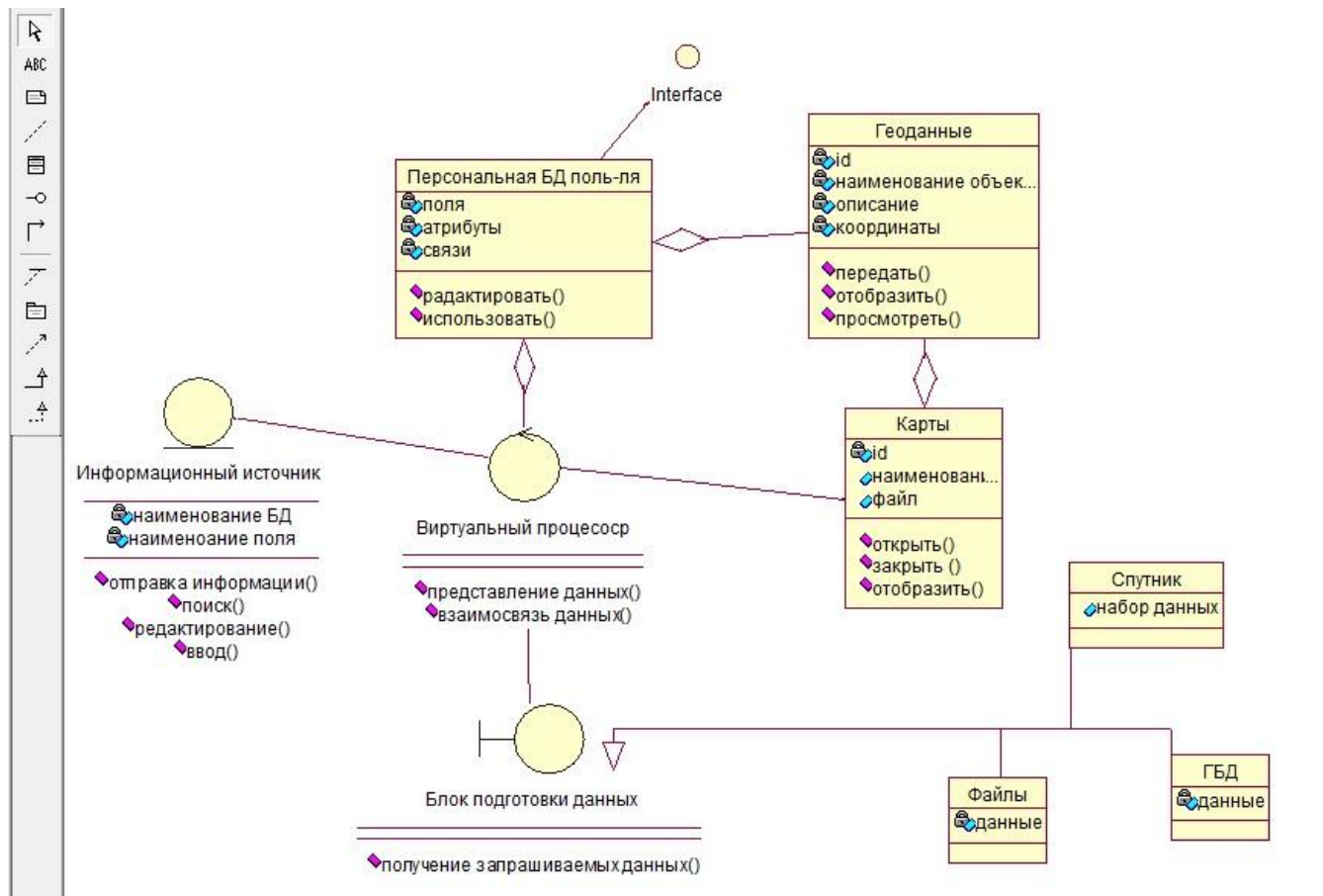


Рисунок 3.2.4 – Диаграмма классов

Диаграмма классов реализуемой геоинформационной системы представляется в виде взаимодействующих представлений об организации потоков событий в системе. В данном случае была необходимость в создании граничного класса («boundary») с наименованием «Блок подготовки данных», который связан отношением «агрегация» с такими классами как «Спутник», «ГБД», «Файлы», что показывает взаимосвязь данных классов типа "часть-целое".

Представление класса как граничного имеет свои обстоятельства, так как именно граничные классы могут выразить положение данного объекта системы. Граничный класс располагается на границе системы с внешней средой и непосредственно взаимодействует с актерами, но не является составной частью системы.

На диаграмме классов представлены помимо граничного ещё два специфических класса, таких как класс-сущность («entity») и управляющий класс («control»). Первый из них представлен в разрабатываемой система как класс «Информационный источник», что характерно для типа данного класса. Класс-сущность содержит информацию, которая должна храниться постоянно и не уничтожается с уничтожением объектов данного класса или прекращением работы моделируемой системы, связанные с выключением системы или завершением программы.

Второй тип класса представлен на диаграмме классом «Виртуальный процессор», так как управляющий класс отвечает за координацию действий других классов. Управляющий класс отвечает за координацию действий других классов. Данный класс является активным и инициирует рассылку множества сообщений другим классам модели.

Неотъемлемым составляющим любой общей диаграммы классов является класс – интерфейс, который служит для спецификации таких элементов модели, которые видимы извне, но их внутренняя структура остается скрытой от клиентов. Интерфейсы не могут содержать ни атрибутов, ни состояний, ни направленных ассоциаций. Они содержат только операции без указания особенностей их реализации. В реализуемой геоинформационной системе таким классом будет служить класс «Браузер».

Все рассмотренные ранее диаграммы отражали концептуальные и логические аспекты построения модели системы. Логическое представление заключается в представлении понятий, которые не имеют материального воплощения. Так, все элементы логического представления, такие как классы, ассоциации, состояния, сообщения, не существуют материально или физически. Они всего лишь отражают понимание статической структуры системы или динамические моменты ее поведения.

Чтобы создать конкретную физическую систему, в данном случае геоинформационную систему «Оценки рисков для ведения хозяйственной деятельности на арктической территории РФ» необходимо перевести все элементы логического представления в конкретные материальные сущности. Для этого существует другой угол зрения на модельное представление системы, а именно - физическое представление модели. При работе с языком UML это означает объединение связанных между собой физических сущностей, которое включает в себя программное и аппаратное обеспечение и персонал.

Окончательный проект создаваемой геоинформационной системы должен состоять из согласованных между собой моделей логического и физического представлений. Для физического представления моделей в структурированном языке UML применяются диаграммы реализации, которые включают в себя две отдельные канонические диаграммы: диаграмму компонентов и диаграмму развертывания.

Диаграмма компонентов, описывает особенности физического представления системы и позволяет определить архитектуру разрабатываемой системы, установив зависимости между программными компонентами. Во многих программах разработки модуль или компонент соответствует файлу, а пунктирные стрелки, соединяющие модули, показывают отношения взаимозависимости. Основными графическими элементами диаграммы компонентов являются компоненты, интерфейсы и зависимости между ними.

При разработке диаграммы компонентов для системы «Оценки рисков для ведения хозяйственной деятельности на арктической территории РФ» были использованы, в том числе и специализированные модули, которые являются частью программной системы и требуют памяти для своего хранения и процессора для исполнения (Рисунок 3.2.5).

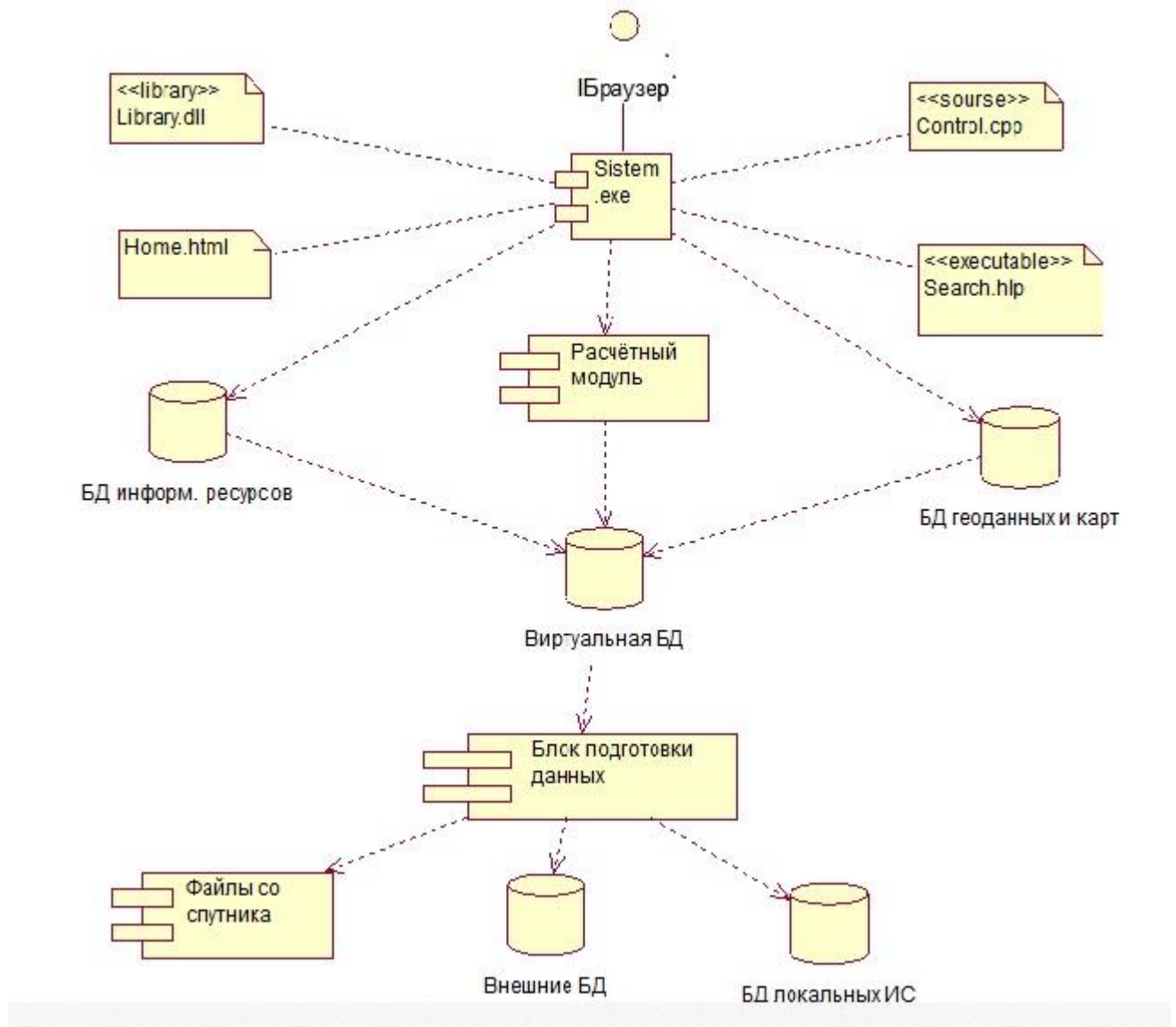


Рисунок 3.2.5 – Диаграмма компонентов

На представленной диаграмме компонентов разрабатываемой системы можно увидеть реально существующий прототип модели системы.

Диаграмма состоит из компонентов, которые представляют собой физически существующую часть системы, которая обеспечивает реализацию классов и отношений, а также функционального поведения моделируемой программной системы. Для наибольшей наглядности в языке UML предусмотрены спецификации различных видов компонентов. На рисунке 3.2.4. они представлены в виде компонентов «Library.dll», «Control.cpp», «Home.html», «Search.hlp», «Файлы со спутника», «Внешние БД». Каждый из них имеет свои спецификации в

зависимости от указанного стереотипа, который отвечает за функциональность данного компонента. Так компонент «Library.dll» имеет стереотип <<library>>, который указывает на то, что этот компонент-файл представляется в форме динамической или статической библиотеки. Компонент «Control.cpp» имеет стереотип <<source>> что означает представление файла с исходным текстом программы, который после компиляции может быть преобразован в исполнимый файл. Компонент «Файлы со спутника» имеет стереотип <<file>>, который представляется в виде произвольного физического файла. Компонент «Внешние БД» представлен со стереотипом <<table>>, который представляется в форме таблицы базы данных. А компонент «Search.hlp» имеет стереотип <<executable>>, который является исполнимым файлом и может выполняться на компьютерной платформе.

На диаграмме компонентов также, как и на диаграмме классов используются интерфейсы. На разработанной диаграмме компонентов для построения системы «Оценки рисков для ведения хозяйственной деятельности на арктической территории РФ» интерфейс представлен в виде компонента «ИБраузер», который реализуется с помощью компонента «System.exe», что представлено в виде зависимости типа «отношения реализации».

При построении диаграммы компонентов участвуют только два вида зависимостей «отношения реализации» и «отношение зависимости». Другими словами, отношения зависимости на диаграмме компонентов является отношениями программного вызова и компиляции между различными видами компонентов, кроме случая между компонентом и интерфейсом, где отношения устанавливаются в виде реализации или использования в качестве процесса своего выполнения компонента-интерфейс.

Физическое представление системы не может быть полным, если отсутствует информация о том, на какой платформе и на каких вычислительных средствах она реализована. Если создается простая программа, которая может выполняться локально на компьютере пользователя, не используя никаких распределенных устройств и сетевых ресурсов, то необходимости в разработке дополнительных диаграмм нет. Однако при создании корпоративных или распределенных приложений требуется визуализировать сетевую инфраструктуру программной системы [67].

В данном случае создаётся сложная геоинформационная система, которая будет реализована в сетевом варианте, на различных вычислительных платформах и технологиях доступа к распределенным базам данных, с использованием сети Интернет. Это определяет необходимость решения следующих задач при проектировании: обеспечение безопасности и устойчивости доступа к информации. Их выполнение зависит также и от физического размещения узлов системы, таких как серверы, рабочие станции, каналы связи и хранилища данных. С целью определения программных и технологических особенностей реализации

распределенных архитектур необходимо визуальное представление этих моментов, что и позволяет сделать UML, с помощью построения диаграммы развертывания, которая является второй формой физического представления системы.

Она необходима только при проектировании сложных информационных систем, так как служит для описания общей конфигурации и топологии распределенной системы, а также располагает изображения компонентов по отдельным узлам.

Диаграмма развертывания включает в себя графические изображения процессоров, процессов, устройств и существующих связей между ними. Она строится в одном экземпляре для всей системы в целом и завершает процесс объектно-ориентированного анализа и проектирования для конкретной информационной системы.

Для создания разрабатываемой геоинформационной системы была построена диаграмма развертывания на примере одного клиента (Рисунок 3.2.6)

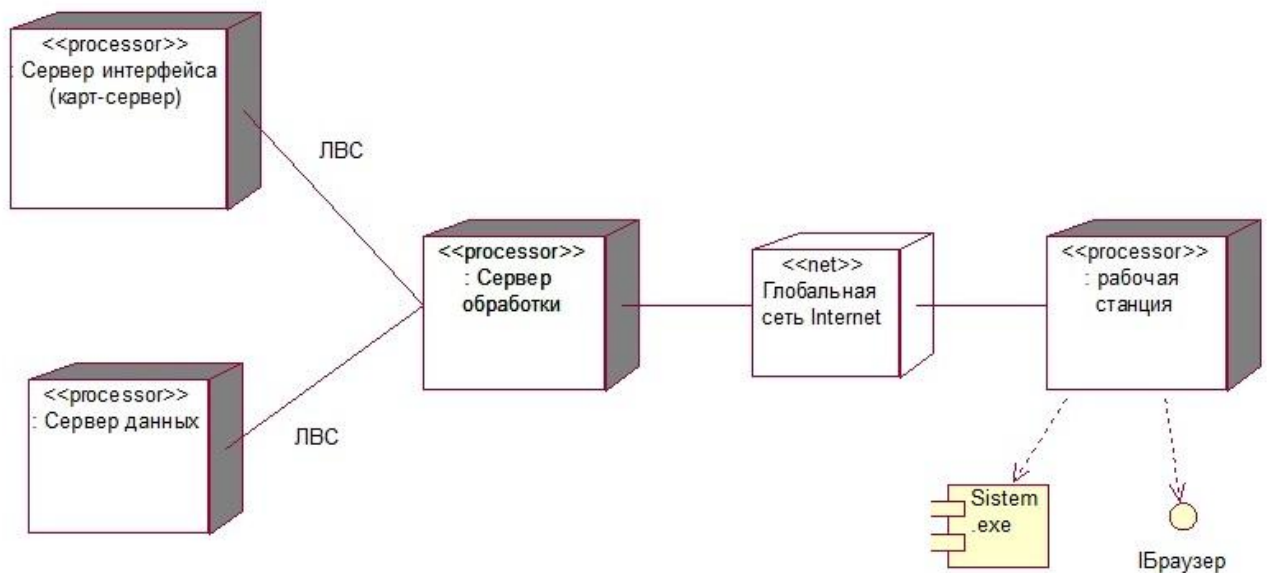


Рисунок 3.2.6 – Диаграмма развертывания

Как показано на рисунке ресурсоемкий узел «Рабочая станция», который выступает в качестве «клиента» соединён с ресурсоемким узлом «Сервер обработки» по средствам глобальной сети Интернет. Наличие сплошной линии связи, в данном случае, указывает на необходимость организации физического канала для обмена информацией между соответствующими узлами.

Кроме соединений на диаграмме развертывания могут присутствовать отношения зависимости между узлом и размещаемыми на нем компонентами, что позволяет видеть размещение физических компонентов указанных на диаграмме компонентов, в тех узлах, где

они реализуются. Так в ресурсоемком узле «Рабочая станция» реализуются компонент с файлом запуска «System» и компонент-интерфейс «Браузер».

3.3 Методика оценки с использованием ГИС

Арктическая зона полна как природными богатствами, так и опасностями.

Для обеспечения хозяйственной деятельности, ведущейся в её пределах необходимо иметь достаточно устойчивый прогноз геоданных о состоянии гидрометеорологических условий в зонах ведения этой деятельности.

Традиционно любой прогноз осуществляется с помощью геофизических моделей, но существует возможность применить к оценки состояния параметров геоданных стохастическую модель, которая положит основу создания распределенной геоинформационной системы на основе разноформатных баз данных. Вся информация, поступающая в данную систему, носит разрозненный характер, а именно хранится в разных форматах и в различных хранилищах, зачастую территориально удаленных друг от друга.

Для анализа возможностей была выбрана база данных отражающая состояние плотности морского дна, созданная в Access 2010 по результатам обработки космической информации предоставленной лабораторией спутниковой океанографии РГГМУ и база данных, которая храниться на сервере лаборатории с представленной там информацией в виде набора текстовой информации в формате . csv . (см. параграф 3.1).

Для создания первичной базы данных в Access было выбрано четыре точки на поверхности морского покрова омывающих Арктическое побережье Евразии, в которых измерялась плотность морского льда. Вся информация была получена при помощи удаленного доступа через программный язык Python.

Представленные данные были выбраны, отсортированы и извлечены за период 2012 - 2013 года по четырем оптимальным точкам.

Данные изначально были перенесены в текстовый формат хранения, предназначенный для представления табличных данных - csv, а затем отправлены в первичную базу данных access, для дальнейшего манипулирования с информацией. База данных была разработана с

целью хранения первичной входной информации о плотности льда в Арктике и постоянного контроля обеспечения целостности данных.

С целью визуализации и дальнейшей работы над полученными данными было реализовано программное приложение в среде разработки Delphi.

Программа с помощью запросов может отражать актуальную информацию о плотности льда за выбранный период времени с 01.01.2012 по 24.08.2013 года в четырех точках, лежащих на протяжении северного морского пути. Каждая точка представляет собой площадь, размером 25 км^2 . Точки расположены в координатах (50,80), (135,85), (173,95), (193,132) (Рисунок 3.3.1).

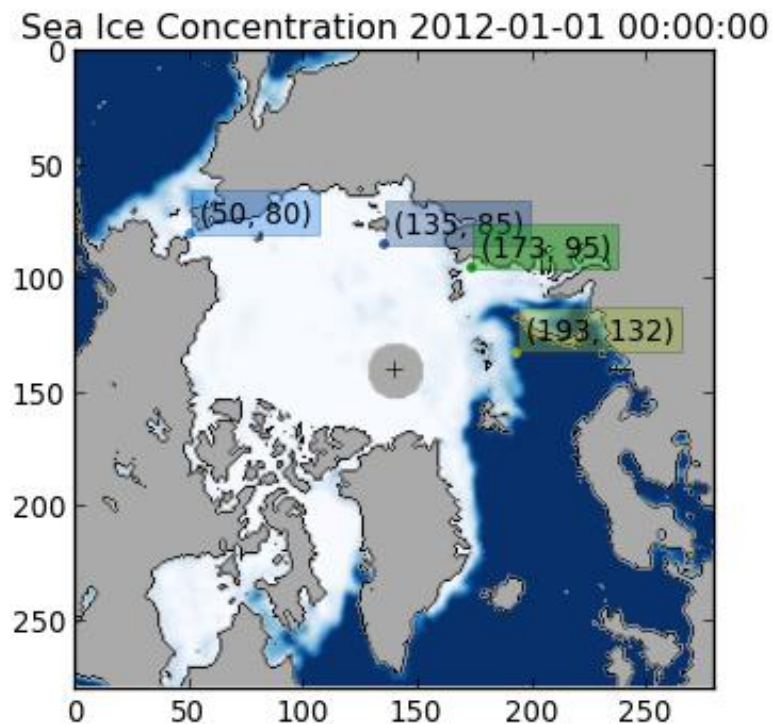


Рисунок 3.3.1 - четыре точки на поверхности Северного ледовитого океана

Каждая точка имеет конкретный набор информации: время сбора данных спутником, показатель плотности льда (Рисунок 3.3.2).

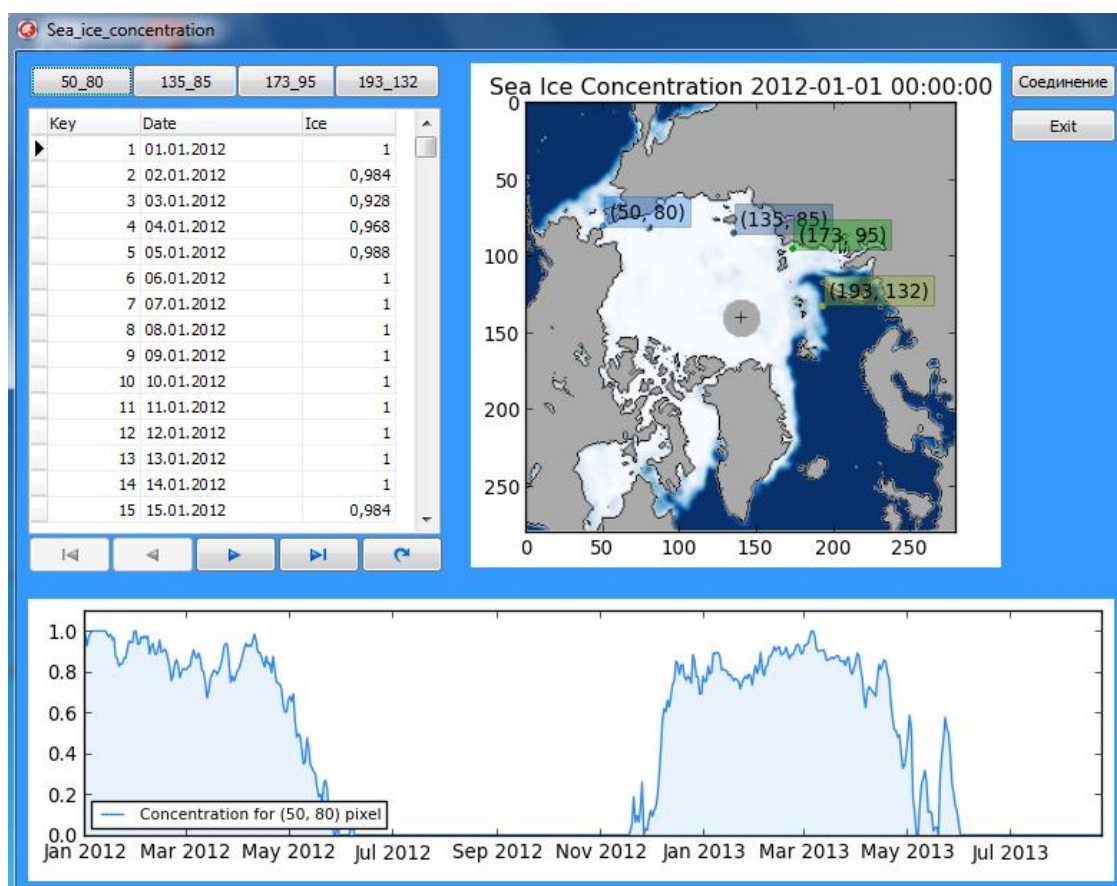


Рисунок 3.3.2 - Показатели плотности льда в каждой выбранной точке представленные в программе Delphi

Время сбора данных спутником составляет каждый день в 00:00.

Показатель плотности льда принимает значение в интервале от 0 – в данной точке отсутствует лед, до 1 – в данной точке плотность льда составляет 100% (Рисунок 3.3.3).

Показатели варьируются в зависимости от времени года, климатических условий, а также выбранной местности на карте.

Объем проводимых манипуляций с данными вынуждает разработчика тратить огромное количество времени на приведение данных к желаемому виду.

Геоинформационная система «Оценки рисков для ведения хозяйственной деятельности на арктической территории РФ» способна по средствам своей организации предоставлять необходимую пользователю информацию из различных источников в течении нескольких минут и производить прогноз по заданным характеристикам.

Для апробации данной системы было выбрана локальная база данных Sea_ice_concentration.accdb и файлы по скорости ветра, представленные в формате данных со спутника WIND_SPEED.csv, хранящиеся на сервере в базе данных лаборатории Спутниковой океанографии. Была создана модель геоинформационной системы, позволяющая получать

информацию из различных источников и проводить прогнозирование состояния выбранных параметров (Рисунок 3.3.4).

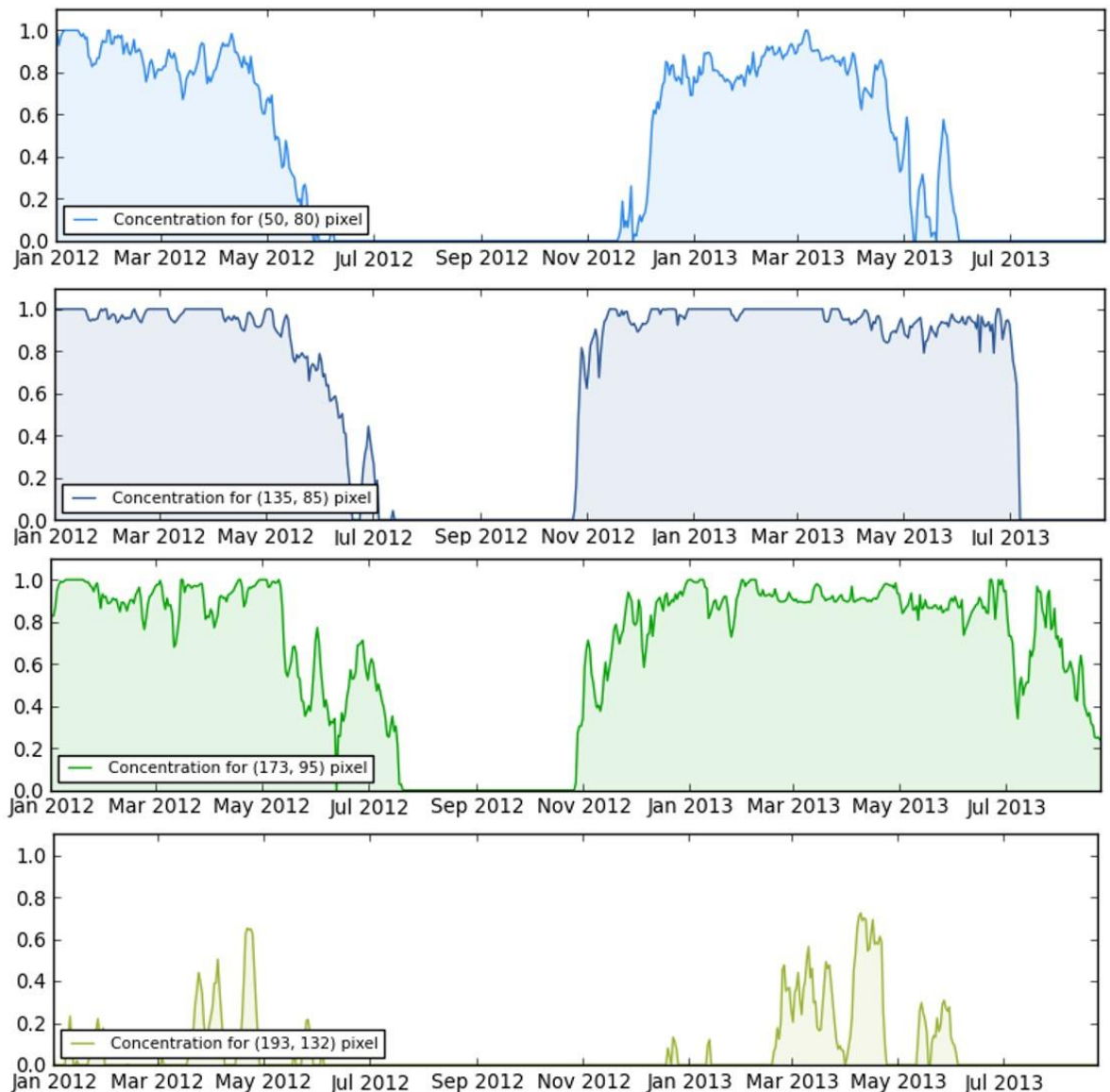


Рисунок 3.3.3 - Диаграмма уровня плотности льда

Для наглядности и предоставления данных о способности системы работать с различными математическими моделями было принято решение создать ГИС с использованием двух параметров: плотность льда и скорость ветра.

В рабочей зоне «Выбор параметров прогнозирования» заносится период времени, на который необходимо создать прогноз, и выбирается регион для которого он осуществляется. В данном случае можно выбрать из четырёх точек, по которым имеются данные и подключены базы к системе. При выборе параметра «Плотность льда» производится прогноз по средствам применения фильтра Калмана – Бьюси и выводится результат в виде графика показаний плотности льда на заданный период времени (Рисунки 3.3.5, 3.3.6).

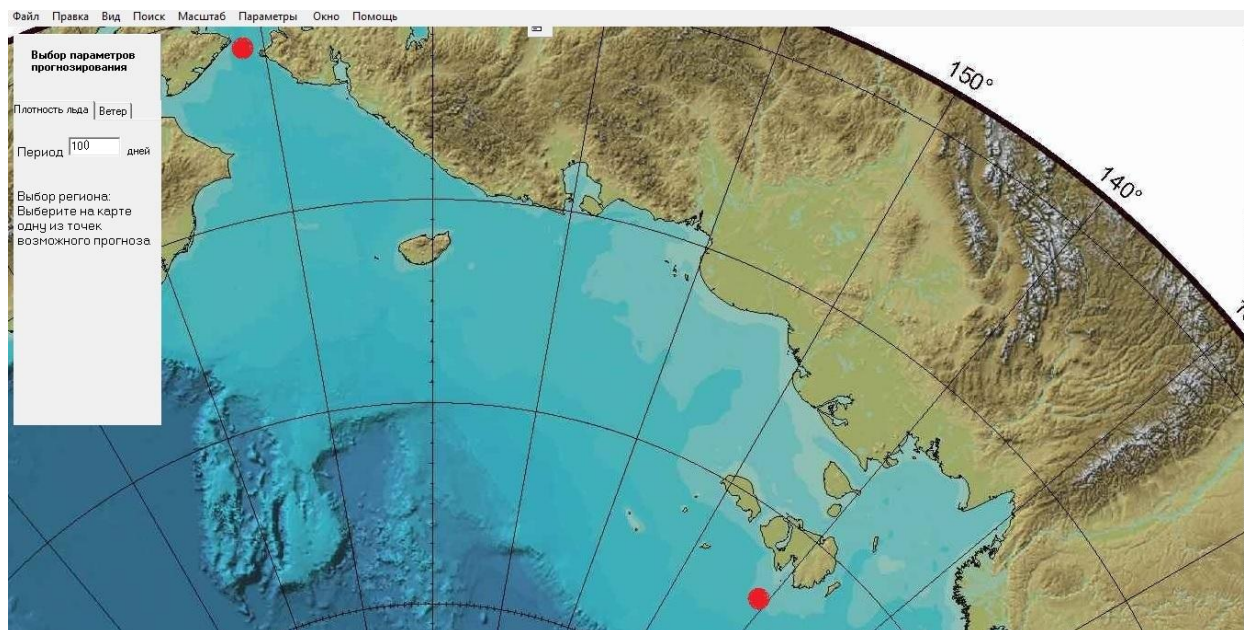


Рисунок - 3.3.4 Выбор параметров прогнозирования состояния природных характеристик в геоинформационной системе

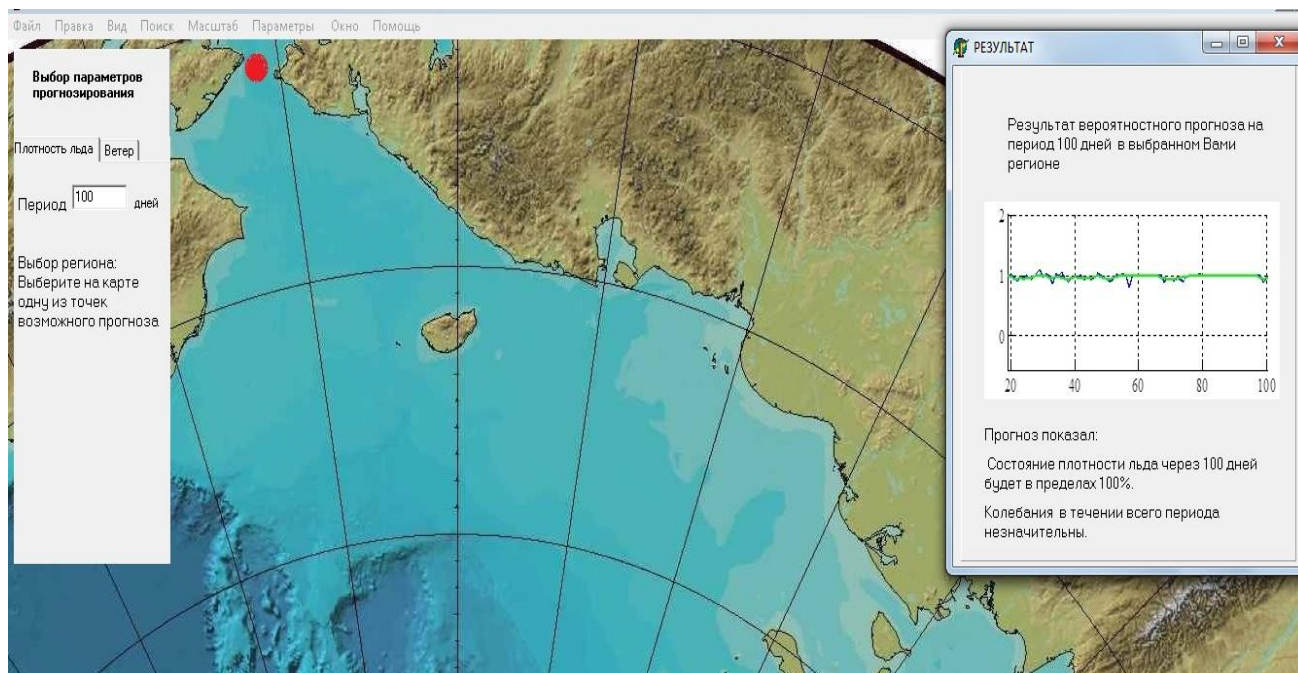


Рисунок 3.3.5 - Общий вид интерфейса программы при выводе результата



Рисунок 3.3.6 - Программный модуль вывода результата

При выборе параметра «Ветер» прогнозирование происходит по средствам математической модели, основанной на статистических характеристиках процессов.

Пользователю необходимо вначале выбрать приемлемый уровень ветра, при котором он может осуществлять хозяйственную деятельность в выбранной точке, тем самым задав верхние и нижние оценки невыхода критерия за которые говорит о состоянии силы ветра в допустимых значениях. После чего необходимо указать период прогноза в днях и нажать клавишу «Enter» (Рисунок 3.3.7).

При выборе силы ветра «Тихий», максимальное приемлемое значение будет находиться в пределах 1,5 м/с, а минимальное, соответственно, в пределах 0,2 м/с. Выход за пределы этих значений будет означать переход в смежный уровень силы ветра, что может повлечь за собой прекращение выполнения работ. После выбора периода происходит расчёт и вывод результатов.

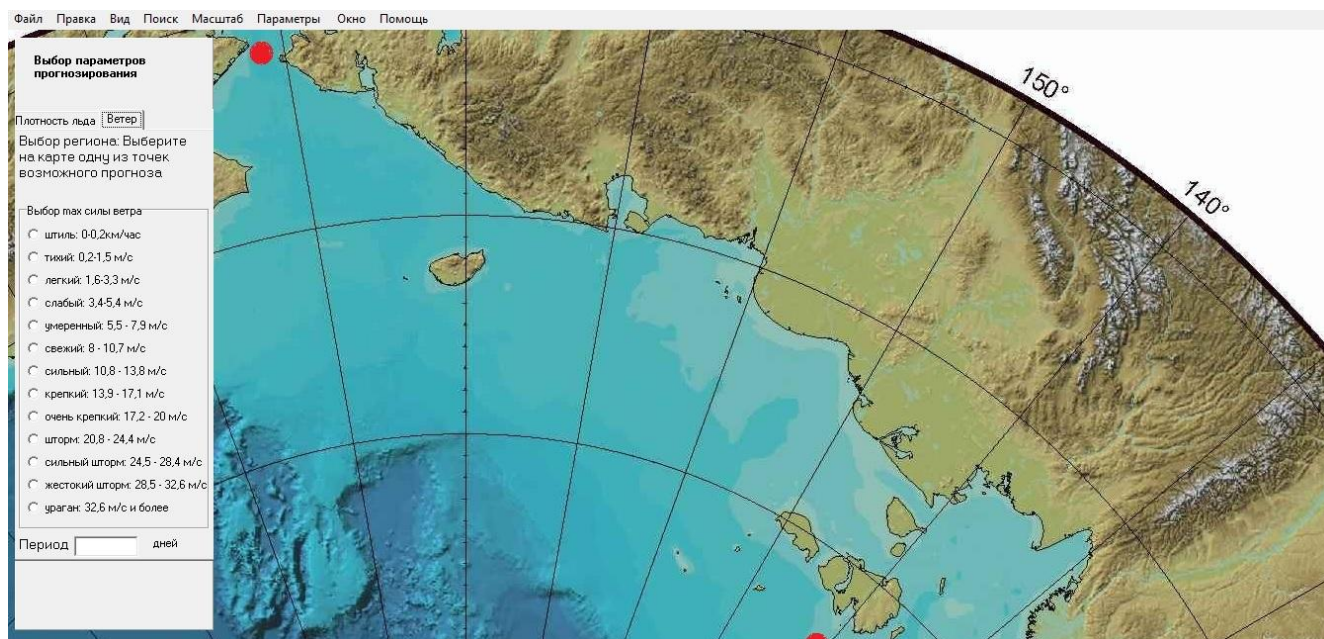


Рисунок 3.3.7 - Общий вид интерфейса программы при выборе параметра «Ветер»

Данная геоинформационная система была построена с помощью императивного, структурированного, объектно-ориентированного языка программирования Borland Delphi 7, а также языков Python, SQL и средств Microsoft Access 2010.

При построении полноценной крупномасштабной геоинформационной системы «Оценки рисков для ведения хозяйственной деятельности на арктической территории РФ» будет возможность подключить большее количество всевозможных баз данных, описывающих различные характеристики природной среды заполярной территории, что увеличит выбор параметров и их показателей и позволит более качественно и безопасно заниматься хозяйственной деятельностью на территории Арктики.

Вывод по главе

В данной главе был проведен анализ входной информации по определенным ранее параметрам и представлена математическая модель обработки гетерогенных данных в рамках верификации геоинформационной системы, которая позволяет одновременно использовать данные для оценок вероятностных характеристик двух параметров (силы ветра и плотности льда), выраженная в схеме взаимосвязи данных.

Для апробации предложенных моделей создания геоинформационной системы на основе распределенных гетерогенных баз данных была построена первичная геоинформационная

система «Оценка климатических параметров для ведения хозяйственной деятельности на арктической территории РФ». Перед реализацией системы было проведено проектирование с использованием методологии объектно-ориентированного анализа, выраженная по средствам визуального языка проектирования UML и программного продукта RationalRose 7.0.

В результате чего были построены концептуальная, статическая и физическая модель сложной системы и представлены в виде диаграмм вариантов использования, классов, компонентов и развертывания. На стадии проектирования были уточнены и скорректированы решения реализации геоинформационной системы, продуманы функциональные возможности системы относительно пользователей и обслуживающего персонала, скорректировано физическое представление систем и размещение отдельных компонентов.

.

ЗАКЛЮЧЕНИЕ

В работе были получены следующие основные результаты:

1. Обоснованы требования к геоданным используемым в прикладных геоинформационных системах с элементами управления, которые получены автором при выполнении НИР «Разработка и развитие методов, моделей и систем геоинформационного управления пространственно-распределенными объектами»;
2. Автором лично проведена классификация современных информационных технологий в области хранения, передачи и обработки геоинформации с использованием вычислительной техники, по критериям распределенности и гетерогенности данных, и выявлено что ни одна представленная технология не может удовлетворить требованиям создания распределённой вычислительной системы обработки геоданных с применением гетерогенных БД;
3. Построена модель геоинформационной системы с учетом препроцессорной подготовки гетерогенных данных, поддерживающая многопользовательский режим и основанная на технологиях доступа к данным как локально, так и через глобальную сеть Интернет. Разработаны принципы и средства, способные эффективно размещать и синхронизировать доступ к пространственным геоданным в различных форматах без явной конвертации. Для объединения и преобразования данных к желаемому виду пользователя, автором было предложено, использовать «виртуальный процессор данных». Как элемент виртуального процессора, «блок подготовки данных» в системе позволяет получить доступ к любым источникам информации, необходимым для принятия управленческого решения;
4. Модифицирована методика проектирования геоинформационной системы на основе распределенных гетерогенных баз данных, которая заключается в объединении технологий создания информационно-управляющей и геоинформационной системы, осложненная гетерогенностью и распределенностью баз данных, входящих в её основу. Автором были предложены: измененная схема, включающая новые шаги и функции проектирования;
5. Разработана математическая модель обработки гетерогенных данных в рамках верификации геоинформационной системы, которая позволяет одновременно обрабатывать данные по двум параметрам (силы ветра и плотности льда), с целью получения оценок вероятностных характеристик данных процессов. Автором лично была предложена схема взаимосвязи различных данных в математической модели.

В диссертации рассматривались данные из разрозненных источников. Всего было представлено 2 параметра, однако предложенные автором научные решения позволяют решать, не только монопараметрические задачи, а также многомерные задачи более чем для двух параметрах и на значительных территориях. Однако это не являлось целью исследования, но задача может быть предметом дальнейших исследований.

Список сокращений и условных обозначений

ГИС	-	географическая информационная система
БД	-	база данных
РИС	-	распределенная информационная система
СУБД	-	система управления базами данных
ГБД	-	гетерогенные базы данных
РБД	-	распределенные базы данных
ГТ	-	глобальная транзакция
ЛТ	-	локальная транзакция
РГГМУ	-	Российский государственный гидрометеорологический университет
ЛВС	-	локальная вычислительная сеть
ОТС	-	организационно-технические системы

СПИСОК ЛИТЕРАТУРЫ

1. Фуфаев Э.В., Разработка и эксплуатация удаленных баз данных: учебник для студ. сред. проф. Образования/ Э.В. Фуфаев, Д.Э. Фуфаев.-М. : Издательский центр «Академия», 2008. – 256 с.
2. Диго С.М., Базы данных: проектирование и использование: Учебник. – М.: Финансы и статистика, 2005. – 592 с.
3. Харрингтон Джен, Разработка баз данных: Пер. с англ. – М.: ДМК Пресс, 2005. – 272 с.
4. ГОСТ 34.321-96 Информационные технологии. Система стандартов по базам данных. Эталонная модель управления данными/ п. 2.1
5. Б.П. Арсеньев, Интегрированные распределённые базы данных/ Б.П. Арсеньев, С.А.Яковлев. – М.:Лань, 2001. – 464 с.
6. Бурковский В.Л., Дорофеев А.Н., Семенин С.В., Моделирование и алгоритмизация управления гетерогенными базами данных в распределенных информационных системах. Воронеж. Гос.техн.университет, 2003. – 136 с.
7. Костенко В.А., Информационные технологии. Основы теории и практики программирования в Internet. В 3 ч. Ч.1. Распределенные системы, динамические библиотеки и СОМ – объекты, связь в распределенных системах, сеть Internet, электронная почта: учебное пособие/ В.А. Костенко. – Орел: Академия ФСО России, 2009. – 244 с.
8. Середович В.А., Геоинформационные системы (Назначение, функции, классификация)
9. Акальцов В.П. Базы данных. В 2-х кн. Книга 2. Распределенные и удаленные базы данных: учебник. – М.: ИД «ФОРУМ»: ИНФА – М, 2009. – 272 с.
10. InfoCity - виртуальный город компьютерной документации [Электронный ресурс]. – Режим доступа: http://www.infocity.kiev.ua/prog/delphi/content/delphi074_2.phtml.
11. Официальный портал органов государственной власти [Электронный ресурс]. Точка доступа: <http://www.eao.ru/?p=1358>.
12. Распределенные системы. Принципы и парадигмы/Э. Таненбаум, М.ван Стеен. – СПб.:Питер, 2003. – 877 с.: ил. – (Серия «Класика computer science»).
13. ГОСТ 34.321-96 Информационные технологии. Система стандартов по базам данных. Издания. – М.:2001. – 24 с.
14. Шаши Шекхар, Санжей Чаула Основы пространственных баз данных./Пер.с англ. – М.:КУДРИЦ-ОБРАЗ, 2004. – 336 с.
15. Колбина О.Н. Разработка и развитие методов, моделей и систем геоинформационного управления пространственно-распределенными объектами: отчет о НИР/ Истомин Е.П.,

- Соколов А.Г., Колбина О.Н./ – Санкт-Петербург: Российский государственный гидрометеорологический университет, 2013. – 102 с.
16. Ивакин А.Я., Шкодырев В.П. Интеллектуальные геоинформационные системы и управление пространственными процессами. Учебное пособие - СПб.: Политехнический университет, 2012.
17. Середович В.А., Геоинформационные системы (назначение, функции, классификация) [Текст]: монография/ В.А. Середович, В.Н. Ключенченко, Н.В. Тимофеева. – Новосибирск: СГГА, 2008. – 192 с.
18. Стас Гераскин, Ник Назаренко История развития технологий публикации геоданных в Интернете [Электронный ресурс]. Ноябрь 1998. Точка доступа: <http://loi.sgcc.ru/gis/inetgis/geowww.html>.
19. Бурковский В.Л., Дорофеев А.Н. Система автоматизированного управления метрологическим обеспечением машиностроительного предприятия// Системы управления и информационные технологии: Сб. науч. тр. Воронеж, 2000. – С.79-86.
20. Гарсия-Молина Г., Ульман Д., Уидом Д. Системы баз данных. Полный курс: Пер. с англ.: Уч. пос. – М.: Издательский дом «Вильямс», 2003. – 1088 с.: ил.
21. Когаловский М.Р. Очерк истории отечественных технологий баз данных // Открытые системы. 2002. №1.
22. Оззу М.Т., Валдуриз П. Распределенные и параллельные системы баз данных // Системы управления базами данных. 1996. №4.
23. Ролланд Ф. Основные концепции баз данных: Пер. с англ.: Уч. пос. – М.: Издательский дом «Вильямс», 2002. – 256 с.: ил.
24. Бурковский В.Л., Дорофеев А.Н., Назаров В.Н. Интеграция гетерогенных баз данных в системы принятия решений // Теория активных систем: Сб. науч. тр. Межд. науч. – практ. конф. Москва, 2001. Том 1. – С.83-84.
25. Калиниченко Л.А. Методы и средства интеграции неоднородных баз данных. – М.: Наука, 1983. – 423 с.: ил.
26. Саймон А. Р. Стратегические технологии баз данных: менеджмент на 2000 год. Пер. с англ./ Под ред. и с предисл. М.Р. Когаловского. – М.: Финансы и статистика, 1999. – 480 с.: ил.
27. Коннолли Т., Бегг К., Страчан А. Базы данных: проектирование, реализация и сопровождение. Теория и практика, 2-е изд.: Пер. с англ.: Уч. пос. – М.: Издательский дом «Вильямс», 2000. – 1120 с.: ил. – Парал. тит. англ.
28. Bright M. W., Hurson A. R., Pakzad S.H. A Taxonomy and current issues in multidatabase systems. Computer, pages 50-60, March 1992.

29. Zisman A., Kramer J. Towards Interoperability in Heterogeneous Database Systems. Imperial college Research Report No. Doc 95/11, December, 1995.
30. Арсеньев Б.П. Интеграция распределенных баз данных [Текст] : учебное пособие / Б. П. Арсеньев, С. А. Яковлев. - СПб. : Лань, 2001. - 461 с.
31. Колбина О.Н. Современные и теоретические аспекты управления распределёнными базами данных. / Истомин Е.П., Колбина О.Н./ Информационные технологии системы: управление, экономика, транспорт, право: Сб.науч.тр./Вып. 1-СПб: ООО «Андреевский издательский дом» - 2011 г.,105 с.
32. Академик. Словари и энциклопедии. [Электронный ресурс]. – Режим доступа: <http://lingvostranovedcheskiy.academic.ru/455/%D0%A1%D0%95%D0%92%D0%95%D0%A0>.
33. Русская Арктика [Электронный ресурс]. – Режим доступа: <http://sevprostor.ru/poleznoe-interesnoe/arctic/542-arctic.html>.
34. Указ президента России от 2 мая 2014 года "О сухопутных территориях Арктической зоны Российской Федерации".
35. Арктика. Настоящее и будущее. Программа международного форума. [Электронный ресурс]. - Режим доступа: <http://atfc.ru/ru/program/> (Дата обращения 15.06.14).
36. Президент России. Молодым ученым и специалистам. Стратегия развития Арктической зоны Российской Федерации и обеспечения национальной безопасности на период до 2020 года. [Электронный ресурс]. - Режим доступа: <http://www.youngscience.ru/pages/main/documents/5124/11484/index.shtml> (Дата обращения 15.06.14).
37. Самарский А.А. Математическое моделирование / А.А. Самарский, А.П. Михайлов. – М.: Физматлит, 2001. – 320 с.
38. Советов Б.Я. Моделирование систем/ Б.Я. Советов, С.А. Яковлев. – М.: Высш.шк., 2005. – 343 с.
39. Дворецкий С.И. Моделирование систем: учебник для студ.высш.учеб.заведений/ С.И. Дворецкий, Ю.Л. Муромцев, В.А. Погонин, А.Г.Схиртладзе. – М.: Издательский центр «Академия», 2009. – 320 с.
40. Алешин Л.И. Информационные технологии: Учебное пособие/ Л.И. Алешин. – М.: Литера, 2008. – 424 с. (серия «современная библиотека», выпуск 35).
41. Семакин И.Г. Информационные системы и модели. Элективный курс: Учебное пособие/ И.Г. Семакин, Е.К.Хеннер. – М.: БИНОМ. Лаборатория знаний, 2005. – 303 с.

42. Слесарева Л.С. Развитие методов геомоделирования и оценки рисков в геосистемах природного характера (на примере наводнений): диссертация на соискание канд. техн. наук: 25.00.35/ Слесарева Людмила Сергеевна. – СПб., 2013. – 174 с.
43. Колбина О.Н. Информационная система прогноза рисков наводнений в Санкт-Петербурге./ Истомин Е.П., Петров Я.А., Слесарева Л.С./ Информационные технологии системы: управление, экономика, транспорт, право: Сб.науч.тр./Вып. 10-СПб: ООО «Андреевский издательский дом» - 2013г.,156 с.
44. Hadzilagos V.A. Theory of Reliability in Database Systems. Journal of the ACM, vol.35, No. 1, 1988, Pg.: 121-145.
45. Bernstein P.A., Hadzilacos V., Goodman N. Concurrency Control and Recovery in Database Systems, Addison-Wesley Series In Computer Science, Addison-Wesley, United States of America, 1987.
46. Дорофеев А.Н., Бурковский В.Л., Анализ сериализуемости глобальных транзакций в распределенной системе мульт-БД // Современные проблемы информатизации в непромышленной сфере и экономике: Сб. науч. тр. VI Международной электронной науч. конф. Воронеж, 2001. – С.89-90.
47. Attaluri G.K., Bradshaw D.P., Coborn N. The CORDS Multidatabase Project. Technical Report. 1995.
48. Когаловский М.Р. Энциклопедия технологий баз данных. М.: «Финансы и статистика», 2002, 800с.
49. Бурковский В.Л., Дорофеев А.Н., Семенин С.В., Моделирование и алгоритмизация управления гетерогенными базами данных в распределенных информационных системах. Воронеж. Гос.техн.университет, 2003. – 136 с.
50. Свидетельство о регистрации №19308 Российская Федерация, Программа расчета георисков и вероятности появления нагонных наводнений /Истомин Е.П., Колбина О.Н., Слесарева Л.С.; заявитель и патентообладатель РГГМУ. – Опубл. 24.06.13.
51. Breitbart Y. Multidatabase Interoperability, Sigmod Record, 19(3):53-60, September 1990.
52. Дорофеев А.Н., Бурковский В.Л., Применение протокола атомарной фиксации в системе мульт-БД // Промышленная автоматика: Межвузовский сборник научных трудов. Воронеж, 2001. – С. 30.
53. Истомин Е.П. Сетевые методы и модели распределения автоматизированных систем: диссертация на соискание докт. техн. наук: 05.13.06/ Истомин Евгений Петрович. – СПб., 1998. – 263 с.
54. ГОСТ 34.003-90 Информационная технология. Комплекс стандартов на автоматизированные системы. Термины и определения.

55. ГОСТ 34.601-90 Информационная технология. Комплекс стандартов на автоматизированные системы. Автоматизированные системы. Стадии создания.
56. ГОСТ 34.602-89 Информационная технология. Комплекс стандартов на автоматизированные системы. Техническое задание на создание автоматизированной системы.
57. ГОСТ 34.603-92 Информационная технология. Виды испытаний автоматизированных систем.
58. ГОСТ 34.201-89 Информационная технология. Комплекс стандартов на автоматизированные системы. Виды, комплектность и обозначение документов при создании автоматизированных систем.
59. РД 50-34.698-90 Автоматизированные системы. Требования к содержанию документов.
60. Емельянова Н.З., Партыка Т.Л., Попов И.И. Основы построения автоматизированных информационных систем: Учебное пособие. – М.: ФОРУМ: ИНФРА-М, 2007. – 416 с.
61. Пат. №2014621110 Российская Федерация, База данных Ice_Concentration [Текст]/Истомин Е.П., Колбина О.Н., Степанов С.Ю., Миранков В.А.; заявитель и патентообладатель РГГМУ. – Заявл. № 2014620482, опубл. 07.08.14.
62. Колбина О.Н. Феномен геоинформационного управления и принципы его реализации / Истомин Е.П., Кирсанов С.А., Соколов А.Г., Колбина О.Н. / – Вестник СПбГУ, серия 7, выпуск 4, СПбГУ, 2014.
63. Духин С.В. Формализация геоданных на основе множественно-реляционной модели // Системы и средства информатики. Специальный выпуск «Математические модели и методы информатики, стохастические технологии и системы». – М.: ИПИ РАН, 2005. С. 253–269.
64. Розенберг И.Н., Духин С.В. Автоматизированная система ведения геоинформационной базы данных, увязанная с параметрами работы и развития ОАО «РЖД» //Труды девятой научно-практической конференции «Информационные технологии в железнодорожном транспорте». «ИНФОТРАНС-2004», 6-9 октября, Санкт-Петербург, 2004. С. 12 – 16.
65. Дулин С.К., Духин С.В. Множественно-реляционная модель как средство обеспечения гомоморфизма геоинформационной системы. – М.: ВЦ РАН, 2006. – 30 с.
66. Колбина О.Н. Сложная информационная система прогнозирования рисков с применением фильтра Калмана – Бьюси / Истомин Е.П., Новиков В.В., Сидоренко А.Ю., Колбина О.Н., Степанов С.Ю. // – Ученые записки РГГМУ, выпуск 36, РГГМУ, 2014.
67. Нотация и семантика языка UML[Электронный ресурс] //Национальный открытый университет «Интуит» - Режим доступа: <http://www.intuit.ru/> (Дата обращения 15.06.14).

ПРИЛОЖЕНИЕ

Приложение А. Скрипт «блока подготовки данных» принятых от системы SOLab

```
from xml.dom import minidom
import os, time
import re
import urllib
import xlswriter

from pydap.client import open_url
import numpy
from numpy import logical_or
import numpy.ma as ma
from scipy import spatial
import datetime

years = range(1981, 2015)

points_list = [(69.195083, 36.455687), (70.053546, 56.100371),
               (73.493434, 69.196073), (76.498797, 88.728638),
               (77.797734, 99.802856), (76.302474, 120.237426),
               (73.212920, 132.604476), (72.957205, 143.722640),
               (70.538282, 169.782209), (66.599172, -169.673357)]

hide_null_values = True
variables = ['wind_speed', 'sst', 'atmosphere_water_vapor_content', 'icecon']
xlsx_path = 'dataArray.xlsx'

# find the nearest element indexes
def do_kdtree(combined_x_y_arrays, points):
    mytree = spatial.cKDTree(combined_x_y_arrays)
    dist, indexes = mytree.query(points)
    return dist, indexes
```

```
def set_coords_value(worksheet, num_row, coords, value=""):
    lat, lon = coords
    worksheet.write('C%d' % num_row, lat)
    worksheet.write('D%d' % num_row, lon)

    worksheet.write('E%d' % num_row, value)

def add_title(worksheet, num_row=1):
    worksheet.write('A%d' % num_row, 'Date')
    worksheet.write('B%d' % num_row, 'Granule name')
    worksheet.write('C%d' % num_row, 'latitude')
    worksheet.write('D%d' % num_row, 'longitude')
    worksheet.write('E%d' % num_row, 'value')

opendap_catalog = 'http://opendap.solab.rshu.ru:8080/opendap/hyrax/allData/'

# wind_speed
def wind_speed_read_parameter_value(dataset, pp_name):
    attrs = dataset[pp_name].attributes.keys()
    fill_value = dataset[pp_name]._FillValue

    if 'scale_factor' in attrs:
        scale_factor = dataset[pp_name].scale_factor
    else:
        scale_factor = 1
    if 'add_offset' in attrs:
        add_offset = dataset[pp_name].add_offset
    else:
        add_offset = 0

    valid_min = dataset[pp_name].valid_min
    valid_max = dataset[pp_name].valid_max

    #read parameter data and remove singleton dimensions:
    data_array = dataset[pp_name][:].squeeze()
```

```
data_array = ma.masked_where(data_array < valid_min, data_array)
data_array = ma.masked_where(data_array > valid_max, data_array)
data_array = ma.masked_where(logical_or(data_array == fill_value,
                                         data_array is None),data_array) * scale_factor + add_offset
return data_array
```

```
def wind_speed_handler(granule, granule_date, worksheet, num_row):
```

```
    try:
        dataset = open_url(granule)
    except KeyboardInterrupt:
        raise KeyboardInterrupt
    except:
        time.sleep(2)
    return num_row
```

```
lat = dataset['lat'][:]
lon = dataset['lon'][:]
```

```
lat_fv = dataset['lat'].FillValue
lon_fv = dataset['lon'].FillValue
```

```
lat = ma.masked_where(lat == lat_fv, lat)
lon = ma.masked_where(lon == lon_fv, lon)
```

```
lat = lat * dataset['lat'].scale_factor
lon = lon * dataset['lon'].scale_factor
#fix longitude:
lon = ma.where(lon <= 180, lon, lon - 360)
```

```
wind_speed = wind_speed_read_parameter_value(dataset, 'wind_speed')
combined_x_y_arrays = numpy.dstack([lat.ravel(),lon.ravel()])[0]
#points_list = list(points.transpose())
dist, indexes = do_kdtree(combined_x_y_arrays,points_list)
```

```
flag_added_granule = False

for i in range(len(points_list)):
    if dist[i] > 0.2:
        if hide_null_values:
            continue
        if not flag_added_granule:
            flag_added_granule = True
            worksheet.write('B%d' % num_row, os.path.basename(granule))
            worksheet.write('A%d' % num_row, granule_date)
            set_coords_value(worksheet, num_row, points_list[i])
            num_row += 1
            # set NULL values
            continue
    index = indexes[i]
    f_lat, f_lon = combined_x_y_arrays[index].tolist()

    print f_lat, f_lon
    # indexes in lat and lon
    ind_mas = numpy.where(numpy.logical_and (lon==f_lon , lat==f_lat))

    x_ind = ind_mas[0][0]
    y_ind = ind_mas[1][0]

    print x_ind, y_ind
    print 'dist: ', dist[i]
    print "wind_speed: ", wind_speed[x_ind, y_ind]
    print 'granule_date: ', granule_date
    value = wind_speed[x_ind, y_ind]
    if type(value) == ma.core.MaskedConstant:
        if hide_null_values:
            continue
        if not flag_added_granule:
            flag_added_granule = True
            worksheet.write('B%d' % num_row, os.path.basename(granule))
```

```
        worksheet.write('A%d' % num_row, granule_date)
        set_coords_value(worksheet, num_row, points_list[i])
        num_row += 1
        # set NULL values
        continue

    if not flag_added_granule:
        flag_added_granule = True
        worksheet.write('B%d' % num_row, os.path.basename(granule))
        worksheet.write('A%d' % num_row, granule_date)
        set_coords_value(worksheet, num_row, points_list[i], wind_speed[x_ind, y_ind])
        num_row += 1

del lat
del lon
del wind_speed
del dataset
return num_row

def wind_speed_workflow(worksheet, num_row):
    worksheet.write('A%d' % num_row, 'wind_speed')
    num_row+=1

    catalog =.opendap_catalog + 'ASCAT-L2-12km'
    catalogxml = os.path.join(catalog, 'catalog.xml')

    xmldoc = minidom.parse(urllib.urlopen(catalogxml))
    itemlist = xmldoc.getElementsByTagName('thredds:catalogRef')

    for y in itemlist:
        year_name = y.attributes['name'].value

        if not year_name.isdigit():
            continue

        if not int(year_name) in years:
```

continue

```
print '\n\nStart year: %s' %year_name
year_cat = os.path.join(catalog, year_name)
year_xml = os.path.join(year_cat, 'catalog.xml')

yeardoc = minidom.parse(urllib.urlopen(year_xml))
days_item_list = yeardoc.getElementsByTagName('thredds:catalogRef')
for day in days_item_list:
    day_name = day.attributes['name'].value
    if not day_name.isdigit():
        continue
    print '\nStart day: %s' %day_name

# write day (date) to file
granule_date = datetime.datetime.strptime('%s %s' %(year_name, day_name),
                                           '%Y %j').strftime('%d.%m.%Y')

day_cat = os.path.join(year_cat, day_name)
day_xml = os.path.join(day_cat, 'catalog.xml')
try:
    daydoc = minidom.parse(urllib.urlopen(day_xml))
except:
    pass
datasets_item_list = daydoc.getElementsByTagName('thredds:dataset')

for _dataset in datasets_item_list:

    dataset_name = _dataset.attributes['name'].value
    if dataset_name.startswith('ascat') and \
        dataset_name.endswith('.nc.gz'):
        granule_path = os.path.join(day_cat, dataset_name)
        print 'Start granule: %s' %granule_path

    num_row = wind_speed_handler(granule_path, granule_date,
```

worksheet, num_row)

SST

```
def sst_read_parameter_value(dataset, pp_name):
```

```
    fill_value = dataset[pp_name]._FillValue
```

```
    scale_factor = dataset[pp_name].scale_factor
```

```
    add_offset = dataset[pp_name].add_offset
```

```
    valid_min = dataset[pp_name].valid_min
```

```
    valid_max = dataset[pp_name].valid_max
```

```
    #read parameter data and remove singleton dimensions:
```

```
    data_array = dataset[pp_name][pp_name][:].squeeze()
```

```
    data_array = ma.masked_where(data_array < valid_min, data_array)
```

```
    data_array = ma.masked_where(data_array > valid_max, data_array)
```

```
    data_array = ma.masked_where(logical_or(data_array == fill_value,  
                                             data_array is None),data_array) * scale_factor + add_offset
```

```
    return data_array
```

```
def sst_handler(granule, granule_date, worksheet, num_row):
```

```
    try:
```

```
        dataset = open_url(granule)
```

```
    except KeyboardInterrupt:
```

```
        raise KeyboardInterrupt
```

```
    except:
```

```
        time.sleep(2)
```

```
        return num_row
```

```
    lat = dataset['lat'][:]
```

```
    lon = dataset['lon'][:]
```

```
    sst = sst_read_parameter_value(dataset, 'sst')
```

```
flag_added_granule = False
```

```
for i in range(len(points_list)):
```

```
    lat_coord, lon_coord = points_list[i]
```

```
    if lon_coord < 0:
```

```
        lon_coord += 360
```

```
    ind_lat = (numpy.abs(lat-lat_coord)).argmin()
```

```
    ind_lon = (numpy.abs(lon-lon_coord)).argmin()
```

```
    value = sst[ind_lat, ind_lon]
```

```
    if type(value) == ma.core.MaskedConstant:
```

```
        if hide_null_values:
```

```
            continue
```

```
        if not flag_added_granule:
```

```
            flag_added_granule = True
```

```
            worksheet.write('B%d' % num_row, os.path.basename(granule))
```

```
            worksheet.write('A%d' % num_row, granule_date)
```

```
            set_coords_value(worksheet, num_row, points_list[i])
```

```
            num_row += 1
```

```
            # set NULL values
```

```
            continue
```

```
    print "sst: ", sst[ind_lat, ind_lon]
```

```
    print 'granule_date: ', granule_date
```

```
    if not flag_added_granule:
```

```
        flag_added_granule = True
```

```
        worksheet.write('B%d' % num_row, os.path.basename(granule))
```

```
        worksheet.write('A%d' % num_row, granule_date)
```

```
        set_coords_value(worksheet, num_row, points_list[i], sst[ind_lat, ind_lon])
```

```
        num_row += 1
```

```
del lat
```

```
del lon
```

```
del sst
del dataset
return num_row

def sst_workflow(worksheet, num_row):
    worksheet.write('A%d' % num_row, 'sst')
    num_row+=1

    catalog = opendap_catalog + 'OISST-AVHRR-V2'
    sub_dir = 'AVHRR'
    catalogxml = os.path.join(catalog, 'catalog.xml')

    xmldoc = minidom.parse(urllib.urlopen(catalogxml))
    itemlist = xmldoc.getElementsByTagName('thredds:catalogRef')

    prog = re.compile(r'(\d{8})')

    for y in itemlist:
        year_name = y.attributes['name'].value

        if not year_name.isdigit():
            continue
        if not int(year_name) in years:
            continue

        print '\n\nStart year: %s' %year_name
        year_cat = os.path.join(catalog, year_name, sub_dir)
        year_xml = os.path.join(year_cat, 'catalog.xml')

        yeardoc = minidom.parse(urllib.urlopen(year_xml))
        datasets_item_list = yeardoc.getElementsByTagName('thredds:dataset')

        for t_dataset in datasets_item_list:
            for dataset in t_dataset.getElementsByTagName('thredds:dataset'):
                dataset_name = dataset.attributes['name'].value
```

```
if dataset_name.startswith('avhrr') and (not 'preliminary' in dataset_name) and \
    (dataset_name.endswith('.nc.gz') or dataset_name.endswith('.nc')):
    # print dataset_name
    try:
        dataset_date = prog.findall(dataset_name)[0]
    except:
        continue
    _year = dataset_date[:4]
    _month = dataset_date[4:6]
    _day = dataset_date[-2:]

    granule_date = '%s-%s-%s' % (_year, _month, _day)
    granule_path = os.path.join(year_cat, dataset_name)
    print 'Start granule: %s' % granule_path

    try:
        num_row = sst_handler(granule_path, granule_date,
                               worksheet, num_row)
    except:
        pass

# atmosphere_water_vapor_content
def water_vapor_read_parameter_value(dataset, pp_name):
    data_array = dataset[pp_name][pp_name][:]

    fill_value = 251
    valid_min, valid_max = 0, 250

    data_array = ma.masked_where(data_array > valid_max, data_array)
    data_array = ma.masked_where(data_array < valid_min, data_array)

    scale_factor = dataset[pp_name].attributes['scale_factor']
    add_offset = dataset[pp_name].attributes['add_offset']
```

```
data_array = numpy.float64(ma.masked_values(ma.masked_equal(data_array, None,  
copy=False),
```

```
fill_value, copy=False)) * scale_factor + add_offset
```

```
return data_array
```

```
def water_vapor_handler(granule, granule_date, worksheet, num_row):
```

```
    try:
```

```
        dataset = open_url(granule)
```

```
    except KeyboardInterrupt:
```

```
        raise KeyboardInterrupt
```

```
    except:
```

```
        time.sleep(2)
```

```
        return num_row
```

```
lat = dataset['latitude'][:]
```

```
lon = dataset['longitude'][:]
```

```
water_vapor_all = water_vapor_read_parameter_value(dataset,  
                                                    'atmosphere_water_vapor_content')
```

```
flag_added_granule = False
```

```
# for ascending and descending
```

```
time_v = [0,1]
```

```
for t in time_v:
```

```
    water_vapor = water_vapor_all[t,:,:]
```

```
for i in range(len(points_list)):
```

```
    lat_coord, lon_coord = points_list[i]
```

```
    if lon_coord < 0:
```

```
        lon_coord += 360
```

```
    ind_lat = (numpy.abs(lat-lat_coord)).argmin()
```

```
    ind_lon = (numpy.abs(lon-lon_coord)).argmin()
```

```
value = water_vapor[ind_lat, ind_lon]

if type(value) == ma.core.MaskedConstant:
    if hide_null_values:
        continue
    if not flag_added_granule:
        flag_added_granule = True
        worksheet.write('B%d' % num_row, os.path.basename(granule))
    worksheet.write('A%d' % num_row, granule_date)
    set_coords_value(worksheet, num_row, points_list[i])
    num_row += 1
    # set NULL values
    continue

print "water_vapor: ", water_vapor[ind_lat, ind_lon]
print 'granule_date: ', granule_date
if not flag_added_granule:
    flag_added_granule = True
    worksheet.write('B%d' % num_row, os.path.basename(granule))
    worksheet.write('A%d' % num_row, granule_date)
    set_coords_value(worksheet, num_row, points_list[i], water_vapor[ind_lat, ind_lon])
    num_row += 1
del water_vapor

del lat
del lon
del water_vapor_all
del dataset
return num_row

def water_vapor_workflow(worksheet, num_row):
    worksheet.write('A%d' % num_row, 'water_vapor')
    num_row+=1
```

```
catalog = opendir_catalog + 'SSMI_NC/f16/daily/data'
catalogxml = os.path.join(catalog, 'catalog.xml')

xmldoc = minidom.parse(urllib.urlopen(catalogxml))
years_list = xmldoc.getElementsByTagName('thredds:catalogRef')

dataset_starts_with = 'f16_ssmi'

for year in years_list:
    year_name = year.attributes['name'].value

    if not year_name.isdigit():
        continue
    if not int(year_name) in years:
        continue

    print '\n\nStart year: %s' %year_name
    year_cat = os.path.join(catalog, year_name)
    year_xml = os.path.join(year_cat, 'catalog.xml')

    yeardoc = minidom.parse(urllib.urlopen(year_xml))
    datasets_item_list = yeardoc.getElementsByTagName('thredds:dataset')

    for _dataset in datasets_item_list:

        dataset_name = _dataset.attributes['name'].value
        if dataset_name.startswith(dataset_starts_with) and \
            (dataset_name.endswith('.nc') or dataset_name.endswith('.nc')):

            strip_count = 9
            if dataset_name.startswith(dataset_starts_with + 's'):
                strip_count = 10
            dataset_date = dataset_name[strip_count:][:8]
            _year = dataset_date[:4]
            _month = dataset_date[4:6]
```

```
_day = dataset_date[-2:]

granule_date = '%s-%s-%s' %(_year, _month, _day)
granule_path = os.path.join(year_cat, dataset_name)
print 'Start granule: %s' %granule_path

try:
    num_row = water_vapor_handler(granule_path, granule_date,
                                   worksheet, num_row)
except:
    pass

# ICE
def ice_read_parameter_value(dataset, pp_name):
    fill_value = dataset[pp_name]._FillValue

    scale_factor = dataset[pp_name].scale_factor

    #read parameter data and remove singleton dimensions:
    data_array = dataset[pp_name][:].squeeze()

    data_array = ma.masked_where(logical_or(data_array == fill_value,
                                             data_array is None),data_array) * scale_factor
    return data_array

def ice_handler(granule, granule_date, worksheet, num_row):
    try:
        dataset = open_url(granule)
    except KeyboardInterrupt:
        raise KeyboardInterrupt
    except:
        time.sleep(2)
    return num_row

lat = dataset['latitude'][:]
```

```
lon = dataset['longitude'][:]  
  
lat = lat * dataset['latitude'].scale_factor  
lon = lon * dataset['longitude'].scale_factor  
#fix longitude:  
lon = ma.where(lon <= 180, lon, lon - 360)  
  
ice = ice_read_parameter_value(dataset, 'icecon')  
combined_x_y_arrays = numpy.dstack([lat.ravel(),lon.ravel()])[0]  
#points_list = list(points.transpose())  
dist, indexes = do_kdtree(combined_x_y_arrays,points_list)  
  
flag_added_granule = False  
  
for i in range(len(points_list)):  
    if dist[i] > 0.2:  
        if hide_null_values:  
            continue  
        if not flag_added_granule:  
            flag_added_granule = True  
            worksheet.write('B%d' % num_row, os.path.basename(granule))  
            worksheet.write('A%d' % num_row, granule_date)  
            set_coords_value(worksheet, num_row, points_list[i])  
            num_row += 1  
            # set NULL values  
            continue  
    index = indexes[i]  
    f_lat, f_lon = combined_x_y_arrays[index].tolist()  
  
    print f_lat, f_lon  
    # indexes in lat and lon  
    ind_mas = numpy.where(numpy.logical_and (lon==f_lon , lat==f_lat))  
  
    x_ind = ind_mas[0][0]  
    y_ind = ind_mas[1][0]
```

```
print x_ind, y_ind
print 'dist: ', dist[i]
print "ice: ", ice[x_ind, y_ind]
print 'granule_date: ', granule_date
value = ice[x_ind, y_ind]
if type(value) == ma.core.MaskedConstant:
    if hide_null_values:
        continue
    if not flag_added_granule:
        flag_added_granule = True
        worksheet.write('B%d' % num_row, os.path.basename(granule))
    worksheet.write('A%d' % num_row, granule_date)
    set_coords_value(worksheet, num_row, points_list[i])
    num_row += 1
    # set NULL values
    continue

if not flag_added_granule:
    flag_added_granule = True
    worksheet.write('B%d' % num_row, os.path.basename(granule))
    worksheet.write('A%d' % num_row, granule_date)
    set_coords_value(worksheet, num_row, points_list[i], ice[x_ind, y_ind])
    num_row += 1

del lat
del lon
del ice
del dataset
return num_row

def ice_workflow(worksheet, num_row):
    worksheet.write('A%d' % num_row, 'icecon')
    num_row+=1
    prog = re.compile(r'(\d{8})')
```

```
catalog = opendap_catalog + 'ASI-AMSRE'
catalogxml = os.path.join(catalog, 'catalog.xml')

xmldoc = minidom.parse(urllib.urlopen(catalogxml))
itemlist = xmldoc.getElementsByTagName('thredds:catalogRef')
for y in itemlist:
    year_name = y.attributes['name'].value
    print year_name

    if not year_name.isdigit():
        continue
    if not int(year_name) in years:
        continue

    print '\n\nStart year: %s' %year_name
    year_cat = os.path.join(catalog, year_name)
    year_xml = os.path.join(year_cat, 'catalog.xml')

    yeardoc = minidom.parse(urllib.urlopen(year_xml))
    datasets_item_list = yeardoc.getElementsByTagName('thredds:dataset')
    for dataset in datasets_item_list:
#         for dataset in t_dataset.getElementsByTagName('thredds:dataset'):
            dataset_name = dataset.attributes['name'].value

            if dataset_name.startswith('asi-s') and (dataset_name.endswith('.nc.gz') or
dataset_name.endswith('.nc')):
                try:
                    dataset_date = prog.findall(dataset_name)[0]
                except:
                    continue
                _year = dataset_date[:4]
                _month = dataset_date[4:6]
                _day = dataset_date[-2:]
```

```
granule_date = '%s-%s-%s' % (_year, _month, _day)
granule_path = os.path.join(year_cat, dataset_name)

print 'Start granule: %s' % granule_path
try:
    num_row = ice_handler(granule_path, granule_date,
                          worksheet, num_row)
except:
    pass

if __name__ == '__main__':
    try:
        # create xlsx file
        workbook = xlsxwriter.Workbook(xlsx_path)

        if 'wind_speed' in variables:
            # add list for variable
            worksheet1 = workbook.add_worksheet()
            num_row = 1
            add_title(worksheet1)
            num_row += 1

            wind_speed_workflow(worksheet1, num_row)
            del worksheet1

        if 'sst' in variables:
            worksheet2 = workbook.add_worksheet()
            num_row = 1
            add_title(worksheet2)
            num_row += 1
            sst_workflow(worksheet2, num_row)
            del worksheet2

        if 'atmosphere_water_vapor_content' in variables:
            worksheet3 = workbook.add_worksheet()
```

```
num_row = 1
add_title(worksheet3)
num_row += 1
water_vapor_workflow(worksheet3, num_row)
del worksheet3
```

```
if 'icecon' in variables:
    worksheet4 = workbook.add_worksheet()
    num_row = 1
    add_title(worksheet4)
    num_row += 1
    ice_workflow(worksheet4, num_row)
    del worksheet4
```

```
except KeyboardInterrupt:
    pass
finally:
    workbook.close()
```

Приложение Б. Свидетельство о регистрации электронного ресурса «Программа расчета георисков и вероятности появления нагонного наводнения»



РОССИЙСКАЯ ФЕДЕРАЦИЯ



СВИДЕТЕЛЬСТВО

о государственной регистрации базы данных

№ 2014621110

База данных Ice_Concentration

Правообладатель: *федеральное государственное бюджетное образовательное учреждение высшего профессионального образования «Российский государственный гидрометеорологический университет» (RU)*

Авторы: *Истомин Евгений Петрович (RU), Колбина Ольга Николаевна (RU), Степанов Сергей Юрьевич (RU), Миранков Валерий Александрович (RU)*

Заявка № 2014620482

Дата поступления 17 апреля 2014 г.

Дата государственной регистрации

в Реестре баз данных 07 августа 2014 г.



Руководитель Федеральной службы
по интеллектуальной собственности

Б.П. Симонов

Приложение Г. Свидетельство о регистрации программы для ЭВМ «Прогноз нефтяного разлива»

РОССИЙСКАЯ ФЕДЕРАЦИЯ



СВИДЕТЕЛЬСТВО

о государственной регистрации программы для ЭВМ

№ 2014618853

Прогноз нефтяного разлива

Правообладатель: *федеральное государственное бюджетное образовательное учреждение высшего профессионального образования «Российский государственный гидрометеорологический университет» (РГГМУ) (RU)*

Авторы: *Миранков Валерий Александрович (RU), Колбина Ольга Николаевна (RU), Басов Илья Сергеевич (RU), Бескид Павел Павлович (RU), Истомин Евгений Петрович (RU)*

Заявка № 2014616417

Дата поступления 03 июля 2014 г.

Дата государственной регистрации

в Реестре программ для ЭВМ 29 августа 2014 г.



Врио руководителя Федеральной службы
по интеллектуальной собственности

Л.Л. Кирий