

Министерство образования Российской Федерации
ТОМСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ

Факультет информатики

Кафедра теоретических основ информатики

ДОПУСТИТЬ К ЗАЩИТЕ В ГАК

Зав. кафедрой, д.т.н.

_____ Ю.Л. Костюк

«__» _____ 2003 г.

Антипин Максим Геннадьевич

**СОЗДАНИЕ КОМПОНЕНТЫ ВИЗУАЛЬНОГО ПРЕДСТАВЛЕНИЯ
ПРОСТРАНСТВЕННО-РАСПРЕДЕЛЕННОЙ ИНФОРМАЦИИ В
РАМКАХ ИНТЕРНЕТ ГИС ПОРТАЛА**

Дипломная работа

Научный руководитель,

н.с. ЛГИТ ИОМ СО РАН

_____ П.Б. Милованцев

Исполнитель,

студент гр. 1481

_____ М.Г. Антипин

Электронная версия дипломной работы помещена

в электронную библиотеку. Файл

Администратор

Томск – 2003

Реферат

Дипломная работа 72 с., 14 рис., 53 табл., 26 источников, 4 прил.

ГЕОИНФОРМАЦИОННЫЕ СИСТЕМЫ, LINUX, C++, ИНТЕРФЕЙС, CGI, ИНТЕРНЕТ, PERL, ОБЪЕКТНО-ОРИЕНТИРОВАННОЕ ПРОГРАММИРОВАНИЕ, БИБЛИОТЕКА КЛАССОВ, БИБЛИОТЕКА ФУНКЦИЙ, ESRI, ARCGIS, MAPINFO, MAPXTREME, SWIG.

Объект исследования – алгоритмы и механизмы визуализации пространственно-распределенной информации.

Цель работы – создание приложения для графического просмотра и редактирования пространственно-распределенной информации

Методы исследования – теоретический и экспериментальный (на ЭВМ)

Проведен анализ имеющихся в наличии пространственно-распределенных данных, разработана архитектура приложения согласно техническим требованиям, создана библиотека функций, реализующих функции приложения, создан объектно-ориентированный интерфейс к библиотеке функций в виде библиотеки классов языка Perl, на основе библиотеки классов разработано CGI приложение, удовлетворяющее техническим требованиям.

Содержание

РЕФЕРАТ	2
СОДЕРЖАНИЕ	3
ВВЕДЕНИЕ	5
1 ИНТЕРНЕТ ГИС ПОРТАЛ	7
2 ТЕХНИЧЕСКИЕ ТРЕБОВАНИЯ	9
2.1 ОБЩИЕ ТРЕБОВАНИЯ	9
2.2 ТРЕБОВАНИЯ К ВОЗМОЖНОСТЯМ ПОЛЬЗОВАТЕЛЬСКОГО ИНТЕРФЕЙСА	10
2.2.1 Инструмент выбора слоев для отображения	10
2.2.2 Инструмент для перемещения текущего центра региона	10
2.2.3 Инструмент для масштабирования изображения	10
2.2.4 Миникарта	10
2.2.5 Инструмент выбора элементов слоя	10
2.3 ЭКСПОРТ ДАННЫХ	12
2.4 ФОРМАТ ВХОДНЫХ ДАННЫХ	13
3 СРАВНЕНИЕ С СУЩЕСТВУЮЩИМИ АНАЛОГАМИ	14
3.1 ESRI ArcGIS	15
3.2 MAPINFO	17
3.3 АНАЛИЗ	19
4 ОПИСАНИЕ ПРИЛОЖЕНИЯ	20
4.1 ОБЩЕЕ ОПИСАНИЕ	20
4.2 СИСТЕМА ХРАНЕНИЯ ПРОСТРАНСТВЕННОЙ ИНФОРМАЦИИ И МЕТАДАННЫХ	21
4.2.1 Общее описание	21
4.2.2 ESRI Shapefile	22
4.2.3 Файл описания карты	24
4.3 СИСТЕМА ВИЗУАЛИЗАЦИИ ХРАНИМОЙ ПРОСТРАНСТВЕННОЙ ИНФОРМАЦИИ	26
4.3.1 Общие сведения	26
4.3.2 ImageMagick	27
4.4 СИСТЕМА ОРГАНИЗАЦИИ ИНТЕРФЕЙСА ПОЛЬЗОВАТЕЛЯ	30
4.4.1 Общая информация	30
4.5 РАЗРАБОТКА CGI ПРИЛОЖЕНИЙ	31
4.5.1 Обзор CGI интерфейса	31
4.5.2 Проектирование и разработка CGI приложений	31
4.5.3 Обзор CGI Core	32
4.6 АРХИТЕКТУРА ПРИЛОЖЕНИЯ	35
ЗАКЛЮЧЕНИЕ	38
СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ	39
ПРИЛОЖЕНИЕ А. РУКОВОДСТВО ПОЛЬЗОВАТЕЛЯ	41
A.1 ПОЛЬЗОВАТЕЛЬСКАЯ ЧАСТЬ	41
A.2 АДМИНИСТРАТИВНАЯ ЧАСТЬ	41
ПРИЛОЖЕНИЕ Б. ФОРМАТ ФАЙЛА ОПИСАНИЯ КАРТЫ	44
ПРИЛОЖЕНИЕ В. ОПИСАНИЕ ОБЪЕКТНО-ОРИЕНТИРОВАННОГО ПРОГРАММНОГО ИНТЕРФЕЙСА	51
В.1 ОБЩЕЕ ОПИСАНИЕ	51
В.2 ВСПОМОГАТЕЛЬНЫЕ КЛАССЫ	51
В.3 КЛАССЫ ДЛЯ РАБОТЫ С ФАЙЛАМИ ФОРМАТА ESRI SHAPEFILE	53
В.4 КЛАССЫ ДЛЯ РАБОТЫ С ОБЪЕКТАМИ ФАЙЛА ОПИСАНИЯ КАРТЫ	56
ПРИЛОЖЕНИЕ Г. ИЛЛЮСТРАЦИИ	66
Г.1 ПОЛЬЗОВАТЕЛЬСКАЯ ЧАСТЬ	66

Введение

За последние несколько лет Интернет прошёл путь от занятия для академических кругов до самой современной сферы бизнеса. В значительной мере это было вызвано ростом World Wide Web, где используются графические браузеры, а также высоким интересом со стороны средств массовой информации. Переход от статических HTML страниц к динамическому представлению данных и диалогу с пользователем был осуществлён во многом благодаря введению технологий использования сценариев.

Использование сценариев изменило форму представления информации пользователям Интернет. Кажется, лишь недавно несколько физиков использовали малоизвестный язык со странным названием HTML для установления связи документов в группе серверов CERN. Современный веб-сайт – это не просто директория с текстовыми документами на веб-сервере; он, как правило, включает средство хранения данных и их выборки (обычно реляционные базы данных) и богатый набор средств динамического создания документов (CGI, модули для веб-серверов для интерпретации Perl, PHP, Java, VisualBasic). Сегодня браузер пользователя не просто средство для отображения статических документов содержащих текст и графику. Каждый пользователь может получить доступ к полноценным приложениям, обладающим мощным интерфейсом и обрабатывающим большие объёмы данных лишь имея доступ в Интернет и обыкновенный браузер. Таким образом, веб-серверы из простых хранилищ информации превращаются в серверы приложений.

В то же время бурное развитие получили и GIS-технологии. Постоянно создаются новые и модифицируются существующие версии программных продуктов, связанных с GIS, совершенствуются аппаратные методы дистанционного зондирования, инженерная инфраструктура, ведутся теоретические исследования, связанные с обработкой и визуализацией, а также методологией использования геоинформационных данных. В условиях всеобщего интереса к GIS остро встает вопрос взаимодействия и оперативного обмена информацией между научными организациями, ведущими разработку в сфере геоинформационных технологий либо использующих накопленную географическую информацию в своей научной работе.

Пред лабораторией геоинформационных технологий Института Оптического Мониторинга СО РАН была поставлена задача разработки программного комплекса для интеграции всех имеющихся в распоряжении Института Оптического Мониторинга аппаратных средств получения геоинформационных данных в единую систему хранения, обработки и наглядного представления информации. Программный комплекс должен включать в себя компоненты для работы с аппаратными средствами, единую реляционную базу данных для хранения полученных от аппаратных средств данных и компонент для визуального представления хранимых данных и средств создания топологических и семантических запросов по хранимым данным.

Компонент для визуализации данных и создания запросов было решено выполнить в форме веб-приложения, т.к., во-первых, современные браузеры обладают достаточной функциональностью для создания достаточно сложного с точки зрения интерфейса пользователя приложения. В тоже время веб-серверы позволяют создавать приложения, использующие в своей работе интерфейс с СУБД и средствами протокола TCP/IP, которые необходимы для вышеуказанного программного комплекса. Во-вторых, веб-приложения предъявляют минимум требований к своим клиентам, т.к., по сути, не требуют никаких настроек от клиента, и, более того, полноценные браузеры в настоящий момент распространены на всех основных программно-аппаратных платформах и входят в состав дистрибутивов большинства операционных систем. В-третьих, веб-приложение позволяет решить проблему оперативного обмена информацией, т.к. воспользоваться

сервисами, предоставляемыми веб-приложением, может воспользоваться любой пользователь Интернет.

Создание программного комплекса было поручено группе программистов, каждый из которых отвечает за создание отдельной компоненты. В задачу, данной работы входит создание компоненты визуализации хранимых данных и построения запросов пространственных запросов на основе изображений карт, предоставленных пользователю компонентой.

Проект при поддержке гранта европейского союза INTAS-00-189, и рекомендован к поддержке Институтом Оптического Мониторинга Сибирского Отделения Российской Академии Наук (ИОМ СО РАН) на Лаврентьевском конкурсе молодежных проектов СО РАН. Описание гранта INTAS доступно на сайте Института Оптики Атмосферы. [11]

1 Интернет ГИС портал

Интернет ГИС портал (веб-портал ATMOS) представляет собой набор тематических веб-сайтов, сочетающих классическую мультимедиа подачу информации и онлайн доступ к исследовательским базам данных, моделям и средствам визуализации и анализа. Каждый веб-сайт – это блоки данных, модели и функции визуализации. Блок данных предоставляет собой характеристики физических процессов атмосферы и поверхности земли, полученные в ходе наблюдений и экспериментов. Блок моделей содержит численные модели базовых тематических областей. Система визуализации, используя интернет технологии, предоставляет доступ к информационным ресурсам, графикам и таблицам, диалоговым и геоинформационным системам и сервисным программам.

Проект геоинформационного Интернет портала финансируется Научным Советом Европейского Союза ИНТАС: INTAS 00-189 ATMOS: A scientific WWW Portal for Atmospheric Environment. Проект победил на Лаврентьевском конкурсе молодежных проектов СО РАН и рекомендован к поддержке Институтом Оптического Мониторинга Сибирского Отделения Российской Академии Наук (ИОМ СО РАН). Описание гранта доступно на сайте Института Оптики Атмосферы [11].

Интернет ГИС портал – это информационная система, предоставляющая доступ к географической, метеорологической и климатической информации, в различных формах представления, как для населения, так и для ученых. В портале будут интегрированы следующие тематические сайты:

- Атмосферная химия, спектроскопия и аэрозоли со структурированным представлением измерений аэрозольных оптических характеристик, химических свойств и микрофизики, так же описание процессов атмосферной химии. Кроме того, здесь же для исследователей будет предоставлена спектроскопическая информация по молекулярным атмосферным компонентам с возможностью подсчета распространения излучения в неомогенных атмосферах.
- Радиация, атмосфера и геомагнетизм. Измерения спектров солнечной радиации и анализ солнечного воздействия на атмосферные характеристики.
- Атмосферная радиация. Подсчет потоков радиации в атмосфере, изучение аэрозолей, облаков и воздействия компонент легких газов на радиацию.
- Динамические модели атмосферы, моделям климата и окружающей среды разных уровней от 3D GCM до статистических моделей.
- Сайт оценки качества воздуха и контроля за загрязнением окружающей среды. Сайт будет предоставлять информацию об основных аспектах загрязнения воздуха, оценки степени воздействия на окружающую среду и будет так же содержать интерактивные учебники. Учебные примеры будут взяты из реальных метеорологических, климатических и геоинформационных систем и моделей озера Байкал и Западно-Сибирской равнины, адаптированных к региональной специфике.

Рабочая программа включает в себя реализацию следующего:

1. Дизайн концепции и архитектуры научного тематического веб-портала по атмосферным наукам, представляющего собой распределенную систему сайтов с интерактивными средствами анализа, основанного на интернет технологиях и

предоставляющего мультимедиа информацию, имитационные модели, ГИС и доступ к исследовательским базам данных.

2. Разработка инструментов, структур данных и мета описателей для эффективного хранения, обработки и отображения данных об окружающей среде в интернете.
3. Подготовка и интеграция в портал следующих информационных ресурсов. Базы данных, содержащие результаты атмосферных измерений выполненных исследователями из партнерских институтов Сибири, а так же данные о региональных характеристиках атмосферы и окружающей среды. Систематизация и каталогизация моделей, инструментов, документации, контактов и примеров приложений. Имитационные модели и средства анализа реализованные на выделенных компьютерных центрах для онлайн имитации и статистического анализа атмосферы и поведения ее составляющих. Учебники, онлайн лекции и тесты, ориентированные на студентов и население. Детальное описание и ГИСы Озера Байкал и Западно-Сибирской равнины.
4. Предоставление научному сообществу удаленного доступа к данным и эффективным инструментам анализа для изучения атмосферы и окружающей среды, а так же для студентов и аспирантов в образовательных целях.
5. Связать имитационные инструменты, включающие 3-D климатические, атмосферные модели с региональными ГИС и атмосферными и климатическими данными, с целью поднятия уровня понимания и осведомленности по вопросам окружающей среды в регионе.
6. Предоставление публичным институтам, промышленности и заинтересованной публике ценную информацию, данные и легкие в использование инструменты для анализа окружающей среды. А так же предоставление населению доступных и надежных прогнозов атмосферных и климатических условий.

Разработку возглавляет Институт оптики атмосферы (Томск). По завершению разработки зеркала информационной системы будут созданы в Институте вычислительной математики (Москва), в Британском Центре Атмосферных Данных (the British Atmospheric Data Center (BADC)) в Великобритании, и в Environmental Software and Services в Австрии.

В рамках ГИС портала планируется широкое использование базы данных HITRAN (High-resolution TRANsmissions molecular absorption database), разрабатываемой в Кембридже, США, и, кроме того, интеграция с существующими базами данных и информационными системами [20, 21]. Сам портал разрабатывается на двух языках: на русском и английском. Задача данной работы заключается в создании компоненты ГИС-портала, отвечающей за визуализацию хранимой пространственной информации и предоставление ее в виде графических изображений пользователю. Также в задачу входит создание интерфейса пользователя, позволяющего организовать взаимодействия пользователя с компонентой с целью формирования пространственных запросов на основе предоставленных изображений и управления системой визуализации.

2 Технические требования

2.1 Общие требования

Основным требованием к приложению является поддержка одновременной работы множества пользователей. Это требование связано с тем, что к приложению могут получать доступ одновременно множество пользователей Интернет, и каждый из них должен иметь возможность работать независимо от действий других пользователей. Это требование подразумевает отслеживание сессий пользователей на веб-сервере, минимизацию количества общих для всех пользователей файлов, наличие блокировки в операциях ввода-вывода для совместно используемых файлов, хранение информации, связанной с сессией пользователя в уникальном для каждой сессии хранилище, которым может являться как файл, так и область оперативной памяти.

Следствием основного требования являются повышенные требования к производительности приложения и его емкостной сложности, т.к. количество одновременно работающих с приложением пользователей может достаточно велико для того, чтобы исчерпать ресурсы сервера, на котором запущено приложение. Таким образом, чем меньше использует приложение вычислительных ресурсов сервера, тем больше число одновременно работающих с приложением пользователей. Уменьшения затрат процессорного времени желательно добиваться за счет использования в приложении эффективных алгоритмов и наиболее производительных программных средств разработки.

Также к приложению, как и к любому веб-приложению, предъявляются требования безопасности. И хотя данное приложение не использует в своей работе конфиденциальной информации, средств аутентификации и авторизации пользователей, оно должно иметь средства для проверки данных, вводимых пользователем, как на стороне клиента, так и на стороне сервера на предмет их некорректности или содержания в них потенциально опасной информации. Другими словами, приложение не должно позволять злонамеренному пользователю выполнять на сервере, на котором запущено приложение, непредусмотренные разработчиком действия.

2.2 Требования к возможностям пользовательского интерфейса

Для удобства и эффективности работы пользователя интерфейс приложения должен обеспечивать наличие ряда общепринятых в визуальных средствах GIS инструментов. Работа пользователя с приложением должна выглядеть следующим образом: пользователь изучает изображение региона, применяет на изображении желаемый инструмент (при этом инструмент должен по возможности предоставить географические параметры рабочего региона, которые будут получены после применения инструмента), применяет желаемый инструмент. Такое снижение интерактивности связано со спецификой работы веб-приложения, которое все данные получает по HTTP протоколу, используя сетевые сервисы. Работа этих сервисов очень медленная по сравнению со скоростью работы системы памяти рабочей станции. Поэтому необходимо минимизировать количество получаемых приложением данных и сеансов обмена данными с сервером. Так как каждое применение инструмента влечет за собой возникновение сеанса обмена данными с сервером, необходимо дать пользователю возможность применять инструмент самостоятельно, а не автоматически при работе с изображением. Ниже приведен список инструментов с описанием функциональности.

2.2.1 Инструмент выбора слоев для отображения

Этот инструмент должен предоставлять пользователю список доступных для визуализации векторных слоев. В списке помимо названия слоя, должен присутствовать тип слоя (точечный, содержащий линии, содержащий полигоны), индикатор того, выбран ли слой, и, желательно, значок, демонстрирующий каким именно образом данный слой будет отображаться. Например, для точечных слоев может быть указано изображение одного элемента слоя, для слоев, содержащих линии или полигоны, может быть указан цвет линии, которым будут отражаться элементы данного слоя. Пользователь должен иметь возможность выбирать новые слои или отменять выбор ранее выбранных слоев.

2.2.2 Инструмент для перемещения текущего центра региона

Этот инструмент должен предоставлять пользователю возможность указать новое положение центра региона. При этом инструмент должен указать географические координаты нового центра.

2.2.3 Инструмент для масштабирования изображения

Этот инструмент должен предоставлять пользователю средства для изменения масштаба отображаемого географического региона. Инструмент должен иметь список возможных коэффициентов увеличения или уменьшения масштаба. Инструмент должен также отображать текущие географические координаты региона и географические координаты региона, который будет получен после масштабирования текущего региона.

2.2.4 Миникарта

Для удобства работы с предыдущими двумя инструментами необходима так называемая миникарта. Миникарта предоставляет крупномасштабное изображение всего текущего региона. На этом изображении должен отображаться текущий рабочий регион. Также миникарта может быть использована как инструмент для перемещения центра текущего региона.

2.2.5 Инструмент выбора элементов слоя

Этот инструмент должен позволять пользователю выбирать элементы слоя, находящиеся как в одной точке изображения, так и в некоторой его прямоугольной

области. При этом выбор объектов производится сразу же по всем активным слоям. Результатом применения данного инструмента является страница, на которой отображается атрибутивная информация каждого элемента каждого слоя находящегося в указанной точке или области изображения. Элементы должны быть сгруппированы по типу слоя, к которому они принадлежат. Страница результатов должна иметь возможность быть представленной в форме удобной для печати. Также должна иметься возможность сохранить представленные на странице данные в текстовом файле.

2.3 Экспорт данных

Одной из основных задач приложения является решение проблемы оперативного обмена данными. В связи с этим приложение должно обладать рядом возможностей связанных с экспортом визуальных и атрибутивных данных текущего региона пользователю приложения в различных форматах и различными способами доставки.

Пользователь приложения должен иметь возможность получить копию текущего изображения рабочего региона в одном из распространенных графических форматов. Также, пользователь должен иметь возможность получать атрибутивные данные в текстовом виде, либо представить их в форме, удобной для печати.

В качестве средства доставки сохраняемой информации пользователь должен иметь возможность выбора между немедленной доставкой данных по протоколу HTTP и сохранения их на локальное хранилище данными браузера пользователя, посылкой запрошенных данных пользователю на его адрес электронной почты, либо сохранении их в поддиректории всеобщего доступа на веб-сервере.

2.4 Формат входных данных

В качестве формата входных данных необходимо использовать формат данных GIS-приложения ArcView фирмы ESRI. Это требование продиктовано, прежде всего, распространенностью в Институте Оптического Мониторинга СО РАН данного приложения, что позволяет сохранить имеющийся на данный момент опыт сотрудников, и исключить необходимость в переквалификации.

С другой стороны, данный формат позволяет хранить все виды данных, необходимых для работы приложения. В частности, формат позволяет хранить топологические данные всех необходимых типов (точечные слои, слои линий, слои полигонов). С каждым слоем можно ассоциировать атрибутивные данные, которые хранятся в распространенном формате СУБД dBase (dbf). Распространенность формата и его простота гарантируют легкость создания и доступа к данным как имеющихся, так и будущих слоев регионов. Также это позволит легко экспортировать данные из других компонент программного комплекса и обновлять имеющиеся атрибутивные данные динамически.

Недостатком данного решения является то, что для создания динамически обновляемых слоев регионов необходимо создать программные модули, предоставляющие программисту API для динамического создания и изменения файлов слоев регионов.

3 Сравнение с существующими аналогами

В последнее время интеграция GIS и Интернет становится все более тесной. Каждый производитель крупных коммерческих GIS продуктов включает в свое приложение программные модули для интеграции с Интернет. На данный момент используется два основных подхода к интеграции GIS-приложений и Интернет.

Первый заключается в том, что программные модули GIS-приложения, отвечающие за интеграцию с Интернет, предоставляют клиенту наборы пространственной информации, удовлетворяющей указанному пользователем запросу, без какой-либо интерпретации. Эта информация может быть передана как в формате используемым самим GIS-приложением, так и в более совместимом формате (например, XML). Интерпретацией полученной информацией и, возможно, ее дальнейшей визуализацией занимается клиентское приложение. Такой подход с одной стороны делает модуль интеграции наиболее универсальным, т.к. он может использоваться как обычный источник пространственной информации наряду с файлами на локальных дисках и БД, но с другой стороны требует наличия у пользователя специального программного обеспечения для интерпретации полученных данных.

Второй подход заключается в интерпретации модулем интеграции с Интернет пространственных данных GIS-приложения и выдаче клиентскому приложению данных в ожидаемом им формате. Такими форматами могут быть различные форматы изображений, различные форматы хранения пространственной информации, используемыми различными приложениями, файлы формата XML. Такой подход с одной стороны сильно ограничивает универсальность модуля интеграции GIS-приложения и Интернет и ведет к необходимости создания множества подобных модулей для одного GIS-приложения, но с другой стороны делает это приложение более гибким, масштабируемым, а главное, исключает необходимость создания специального клиентского программного обеспечения для работы со средствами, предоставляемыми GIS-приложением, через Интернет. При этом подходе в качестве клиентского программного обеспечения могут быть использованы стандартные браузеры HTML и XML, которые, на данный момент, доступны практически для любой программно-аппаратной платформы, что делает доступным сервисы GIS-приложения, использующего такой подход, доступными широкому кругу пользователей Интернет.

В существующих на данный момент на рынке приложениях может использоваться комбинация этих двух подходов. Ниже приводится краткое описание модулей интеграции с Интернет двух крупнейших существующих на данный момент коммерческих GIS-приложений.

3.1 ESRI ArcGIS

Этот коммерческий программный продукт американской фирмы ESRI стал стандартом де-факто в России. Его широкому распространению способствовали усилия самой фирмы ESRI [5], которая занимается продвижением своего программного продукта на российском рынке, используя ряд российских партнеров, богатейшая функциональность ArcGIS, которая позволяет решать множество задач, связанных с созданием, редактированием, анализом и выводом различной пространственной информации. Также ArcGIS может быть очень гибко настроена для решения как простых, так и весьма сложных задач, оснащена дополнительными модулями, расширяющими ее базовую функциональность. К существенным недостаткам данной системы можно отнести неоправданно низкую скорость работы, доступность ее только под операционными системами Windows 98/2000/XP.

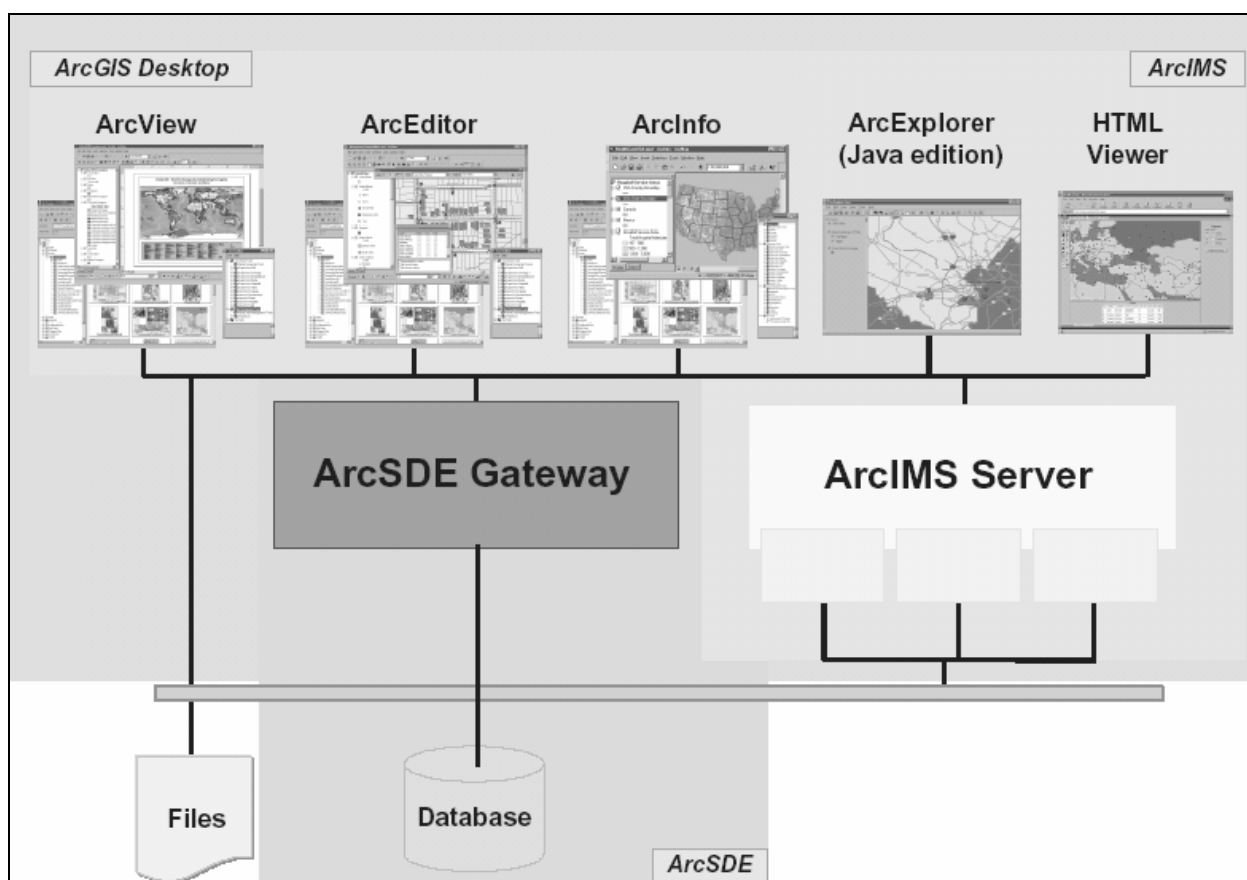


Рисунок 3.1 - Архитектура ArcGIS

Набор модулей для интеграции с Интернет в ArcGIS называется ArcIMS. ArcIMS представляет собой набор серверного и клиентского программного обеспечения. Для взаимодействия клиентской и серверной программных частей используется подмножество XML ArcXML. В качестве клиентского программного обеспечения могут быть использованы ArcView, ArcEditor, ArcExplorer (Java edition), Wireless, а также HTML браузер. Функциональность доступная в HTML браузере существенно ограничена по сравнению с другим возможным клиентским программным обеспечением.

ArcIMS предоставляет клиентскому программному обеспечению следующую функциональность

Сервисы просмотра карт – предоставляет возможность передачи клиентскому приложению пространственной информации в виде изображений в форматах JPEG, PNG, GIF, так и в специальном формате, позволяющем дальнейшую обработку этой

информации на стороне клиента, такую как добавление надписей, дополнительной информации.

Пространственные запросы – предоставляет возможность создавать новые запросы или использовать уже существующие для получения атрибутивной информации об отображаемых данных.

Сохранение данных- можно запросить настоящую пространственную информацию с сервера, который создаст ряд файлов в формате SHP, сожмет их при помощи архиватора и передаст клиентскому приложению.

Геокодирование – позволяет передавать серверу адреса для дальнейшего геокодирования.

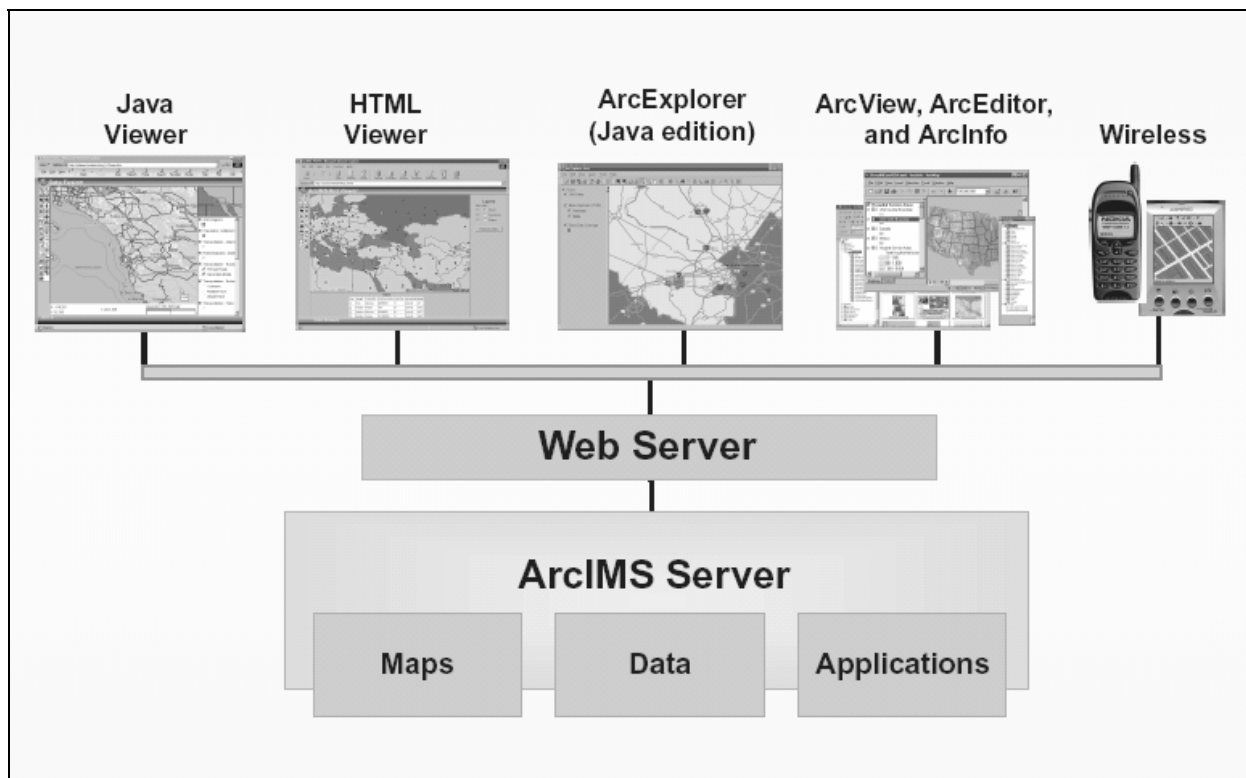


Рисунок 3.2 - Архитектура ArcIMS

Таким образом, набор модулей ArcIMS GIS-приложения ESRI ArcGIS совмещает в себе два вышеописанных подхода, но в основном ориентирован на первый подход.

3.2 MapInfo

Этот коммерческий программный продукт разрабатывается американской фирмой MapInfo Corp [7] и позиционируется как решение для бизнеса, в случае если для более эффективного функционирования бизнес процессов необходимо наглядное представление хранимой пространственной информации.

MapInfo специально разрабатывалась для обработки и анализа информации имеющей адресную или пространственную привязку. В комплект MapInfo входит множество утилит, существенно расширяющих функциональность системы. MapInfo применяется для ведения земельного, лесного кадастра, кадастра недвижимости. Также это коммерческий программный продукт находит применение в следующих областях: градостроительство и архитектура, телекоммуникации, добыча и транспортировка нефти и газа, электрические сети, экология и природопользование, геология и геофизика, железнодорожный и автомобильный транспорт, банковское дело, образование, государственное управление.

Эта система использует для хранения пространственных данных собственный формат. Поддерживает импорт и экспорт векторных данных из большинства векторных форматов. Поддерживает импорт и экспорт изображений из большинства форматов хранения растровых изображений. Поддерживает импорт данных из баз данных, поддерживающих ODBC (Open DataBase Connectivity) [8]. Обладает встроенным языком программирования MapBasic, предоставляет средства для использования функций приложения в языках программирования.

Компонент MapInfo, отвечающий за представление пространственных данных, хранимых MapInfo, в Интернет называется MapXtreme. Эта система состоит из ActiveX компоненты MapX, предназначенной для создания ASP (Active Server Pages) [9] веб-сценариев под управлением IIS (Internet Information Server), приложения MapXBroker, предназначенного для диспетчеризации запросов от MapX к серверам хранения пространственной информации и серверов для хранения пространственной информации MapXServers. Так как MapX представляет собой ActiveX компоненту, предназначенную для включения ее функциональности в сценарии ASP, эта компонента предоставляет множество методов, влияющих на процесс представления пространственных данных в Интернет, что делает эту систему очень гибкой.

На рисунке 3 приведена схема архитектуры система представления пространственной информации системы MapInfo в Интернет.

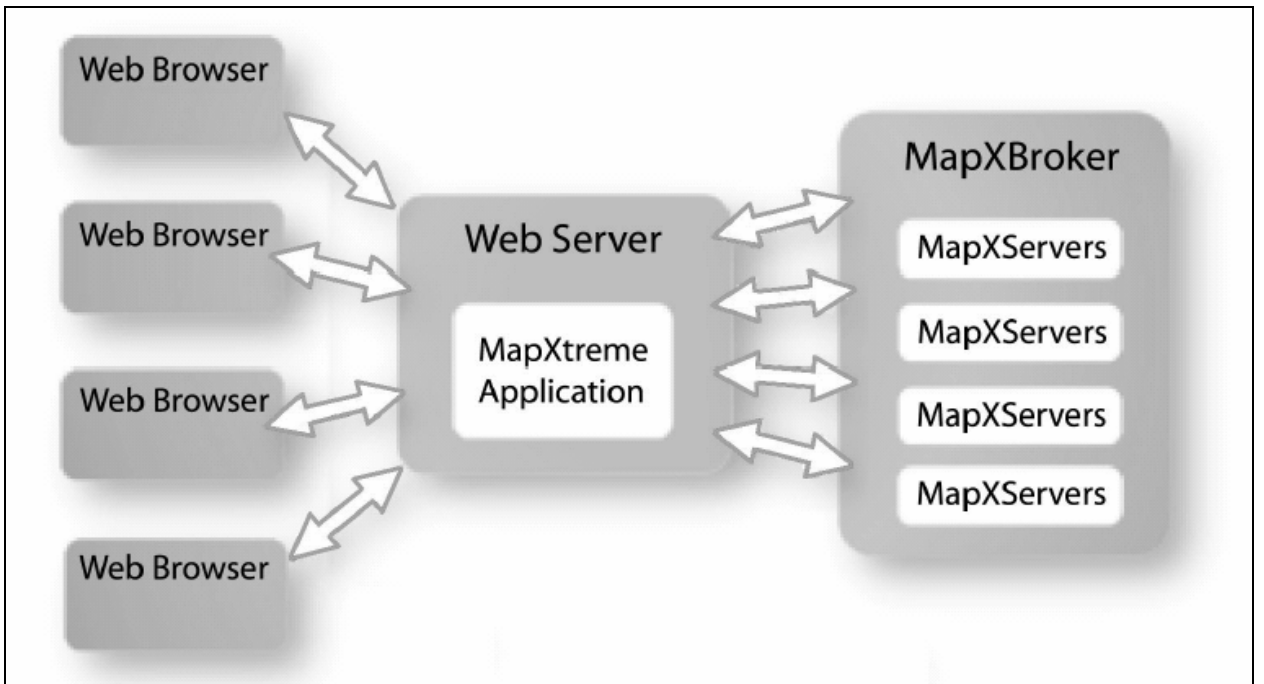


Рисунок 3.3 - Архитектура MapXtreme

3.3 Анализ

Вышеописанные GIS-приложения не удовлетворяют требованиям проекта по ряду причин. ArcGIS и его набор модулей для интеграции с Интернет ArcIMS слишком тяжеловесны, обладают избыточной функциональностью, большая часть которой не находит применения в данном проекте. Также они в целом ориентированы на наличие у пользователя специализированного программного обеспечения, что совершенно неприемлемо в данном проекте. Также ArcGIS предъявляет очень высокие системные требования, что делает затруднительным интенсивную многопользовательскую работу. Кроме того, фирма ESRI выпускает свое программное обеспечение только под операционные системы от Microsoft, такие как Windows 98/2000/XP, что также делает невозможным применение данной системы в данном проекте. Сама система ArcGIS также является платной.

Достоинством системы MapXtreme является ее направленность на представления пространственной информации в Интернет только средствами HTML-браузера клиента. Для этого применяется технология ASP и специальная ActiveX компонента MapX. Такой подход обеспечивает доступность и чрезвычайную гибкость системы представления пространственной информации в Интернет коммерческого приложения MapInfo, но в то же время требует знания языков программирования JScript и VBScript, и ограничивает применение этой системы платформами под управлением коммерческих операционных систем Microsoft Windows 98/2000/XP. Сама система MapInfo также является платной.

На основе анализа функциональности, используемых технологий, системных требований вышеописанных коммерческих продуктов было принято решение создать собственный программный продукт, который бы обладал описанной в требованиях функциональностью, работал бы под UNIX-подобными операционными системами (в частности Linux), не требовал бы от пользователя специализированного клиентского программного обеспечения, использовал бы для своей работы обычный HTML-браузер (Microsoft Internet Explorer, Mozilla, Netscape Navigator), требовал бы от HTML-браузера минимальной поддержки средств создания интерактивных страниц, таких как JavaScript и, по возможности, минимальные системные требования. Таким образом, такое решение позволяет совместить необходимую функциональность с возможностью одновременной работы большого числа клиентов системы, высокой скоростью низкими системными требованиями, отсутствием требований к клиентскому программному обеспечению, доступностью хранимых данных в различных форматах.

4 Описание приложения

4.1 Общее описание

Созданная система визуализации пространственной информации состоит из трех основных частей.

Первая часть – это система хранения пространственной информации. Эта система включает в себя определение форматов для хранения пространственной информации и метаданных, связанных с этой пространственной информацией, способы хранения пространственной информации и метаданных, и набор программных интерфейсов с системой хранения пространственной информации, предоставляющим доступ к функциям сохранения, добавления, изменения хранимой пространственной информации и связанных с ней метаданных, осуществления различного вида поиска по хранимой информации.

Вторая часть – эта система визуализации хранимой пространственной информации. Эта система использует в своей работе функциональность, предоставляемую системой хранения пространственной информации и метаданных. Основной задачей этой системы является создание графического представления пространственной информации на основе описывающих ее метаданных. Также эта система занимается созданием других графических элементов, таких как миникарта, линейка масштаба, легенда. Результатом работы данной системы являются графические файлы в форматах PNG, JPEG.

Третья часть – система организации интерфейса пользователя. Эта система отвечает за представление конечному пользователю информации, интерпретации действий пользователя, организацией запросов к системе хранения пространственной информации и метаданных на основе интерпретации действий пользователя, передачи полученных данных системе визуализации и передачи полученных графических файлов конечному пользователю.

Ниже каждая система описывается подробно, приводится информация об используемых форматах файлов, подробно описываются механизмы работы.

4.2 Система хранения пространственной информации и метаданных

4.2.1 Общее описание

Основными задачами системы хранения пространственной информации и метаданных являются создание, изменение, сохранение пространственной информации, а также связанных с ней метаданных. Также эта система осуществляет поиск пространственной информации, удовлетворяющей различным пространственным запросом. В качестве пространственного запроса могут выступать точка, прямоугольник или полигон. В качестве средства хранения пространственной информации могут быть использованы файлы в формате SHP (Shapefile), используемом в коммерческих продуктах фирмы ESRI, а также произвольная БД, которая поддерживает принятый в UNIX-подобных операционных системах интерфейс DBI (Data Base Independent). Но для полноценного графического представления пространственной информации недостаточно, т.к. она сама по себе не содержит данных, достаточных для ее интерпретации и визуализации. Для создания полноценного графического представления пространственной информации используются файлы описания пространственной информации, которые позволяют объединять множество SHP файлов в карту, задавать тип используемой в карте проекции, единицы измерения координат векторных объектов, хранимых в SHP файлах, способы отображения векторных объектов, принадлежащих определенному SHP файлу, указать способы отображения результатов поиска по векторным объектам принадлежащих определенному SHP файлу.

Система хранения пространственной информации и метаданных имеет следующую внутреннюю структуру. В конфигурационном файле приложения указывается директория для хранения файлов метаданных, в которых хранятся описания карт. Эти файлы имеют расширение *.map*. Директория, в которой хранятся SHP файлы. Для каждой карты, SHP файлы хранятся в поддиректории, имя которой совпадает с именем файла метаданных, описывающим эту карту. Также в конфигурационном файле приложения указывается параметры БД, такие как тип БД, имя БД, имя пользователя, пароль, которая также может служить источником пространственной информации.

Каждый файл или таблица БД, на которую имеется указание в файле описания карты в качестве источника пространственной информации, представляет собой слой карты, который может состоять из точек, линий либо полигонов. Для каждого слоя в файле описания карты могут быть определены параметры, отвечающие за его отображение, такие как цвет элементов слоя, минимальный и максимальный масштаб, при котором отображаются элементы данного слоя, а также фильтры элементов данного слоя. (см. рис. 3)

Ниже подробно описываются элементы системы хранения пространственной информации и метаданных.

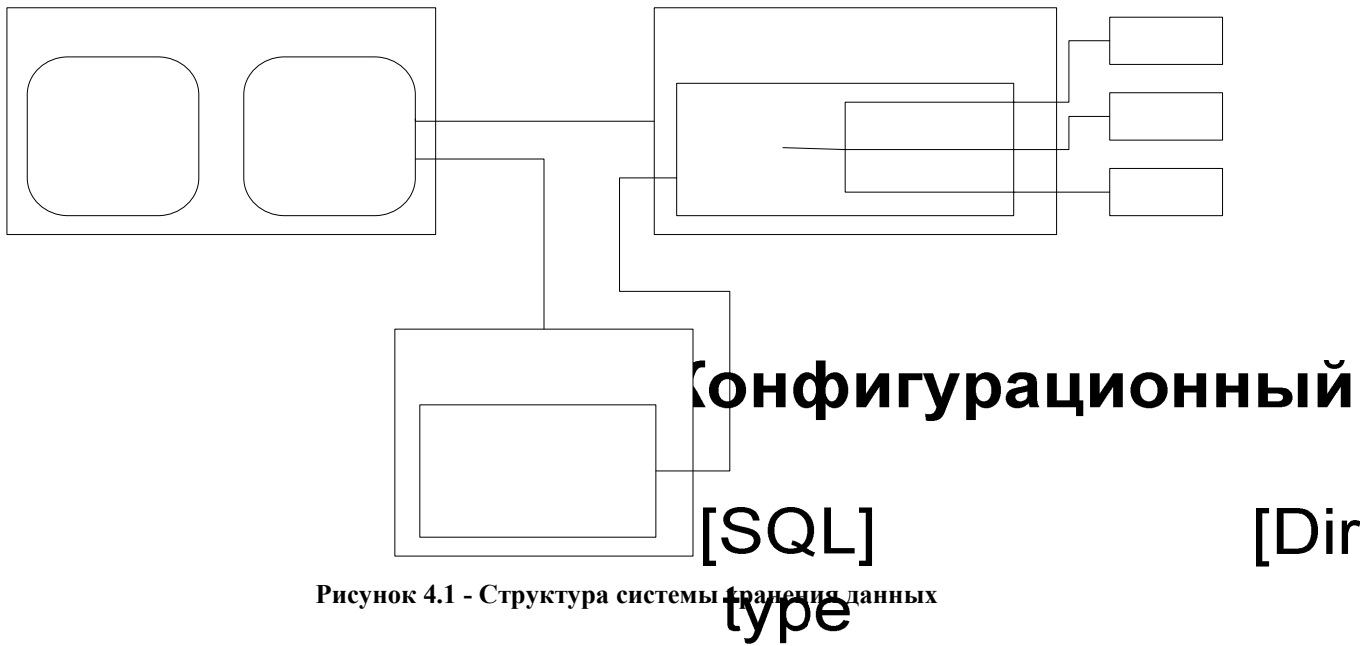


Рисунок 4.1 - Структура системы хранения данных

4.2.2 ESRI Shapefile

Выбор данного формата в качестве основного формата для хранения пространственных данных был обусловлен множеством объективных причин к числу которых можно отнести следующие.

Формат ESRI Shapefile [2] является наиболее полным техническое описание данного формата, предоставленное фирмой ESRI, фирма ESRI не препятствует созданию программного обеспечения для экспорта и импорта в этот и из этого формата

Широкая распространенность продуктов фирмы ESRI в России и за рубежом. Это значительно упрощает процесс адаптации имеющейся на данный момент в распоряжении ИОМ СО РАН пространственной информации для дальнейшего отображения ее при помощи разработанной системы, исключает перекалфикацию сотрудников.

Файлы данного формата содержат как пространственную, так и атрибутивную информацию. При этом пространственная информация хранится без задания определенной топологии. Файлы данного формата могут содержать различные виды геометрических примитивов. Обладают простым форматом и средствами индексирования хранимых объектов, что выгодно сказывается на производительности работы с данными файлами. Для хранения атрибутивной информации используется формат XBase (dBase), для работы с которым доступно большое количество свободно распространяемых библиотек для различных языков программирования высокого уровня.

Shapefile состоит из основного файла, содержащего информацию о геометрических примитивах, индексного файла, содержащего смещения относительно начала основного файла для каждого геометрического примитива, хранимого в основном файле, и файла с атрибутивной информацией формата XBase (dBase). Атрибутивная информация хранимого в основном файле геометрического примитива хранится в строке таблицы, хранимой в файле атрибутивной информации, номер которой соответствует порядковому номеру геометрического примитива в основном файле.

В основном файле используется два типа данных. Это 32-битное целое со знаком, и 64-битное дробное с двойной точностью со знаком. Основной файл имеет заголовок длиной сто байт. Заголовок хранит в себе опознавательную сигнатуру, длину файла, тип хранимых геометрических примитивов, размеры описывающего прямоугольника по трем осям координат, а также ряд зарезервированных полей. Вслед за заголовком идет множество записей, каждая из которых описывает геометрический примитив. Запись,

описывающая геометрический примитив имеет заголовок фиксированной длины, которая зависит от типа геометрического примитива, и содержимого записи произвольной длины.

Таблица 1 – Заголовок файла формата ESRI Shapefile

Позиция	Поле	Значение	Тип
0	Опознавательная сигнатура	9994	Integer
4	Не используется	0	Integer
8	Не используется	0	Integer
12	Не используется	0	Integer
16	Не используется	0	Integer
20	Не используется	0	Integer
24	Длина файла	Длина файла	Integer
28	Версия	1000	Integer
32	Тип геометрического примитива	Тип	Integer
36	Описывающий прямоугольник	Xmin	Double
44	Описывающий прямоугольник	Ymin	Double
52	Описывающий прямоугольник	Xmax	Double
60	Описывающий прямоугольник	Ymax	Double
68	Описывающий прямоугольник	Zmin	Double
76	Описывающий прямоугольник	Zmax	Double
84	Описывающий прямоугольник	Mmin	Double
92	Описывающий прямоугольник	Mmax	Double

Таблица 2 - Типы геометрических примитивов

Значение	Тип геометрического примитива
0	Null shape
1	Point
3	PolyLine
5	Polygon
8	MultiPoint
11	PointZ
13	PolyLineZ
15	PolygonZ
18	MultiPointZ
21	PointM
23	PolyLineM
25	PolygonM

28	MultiPointM
31	MultiPatch

В данном проекте используются только три типа возможных геометрических примитивов. Это Point, Polygon и PolyLine. Следует отметить, что геометрический примитив полигон состоит из замкнутых циклов, не имеющих пересечений друг с другом.

4.2.3 Файл описания карты

Файл описания карты (файл метаданных) используется для объединения множества файлов с пространственными данными в единое целое, задание структуры карты, описания слоев карты, способов их отображения, тип проекции, применяемой для отображения карты. Также в этом файле описываются такие элементы карты как миникарта, линейка масштаба, легенда и прочее.

Файл описания карты имеет расширение *map* и представляет собой текстовый файл, в котором содержатся описания элементов карты. Файл имеет вложенную структуру, т.е. некоторые объекты могут содержать в себе другие объекты. Описания объектов начинаются с имени объекта и заканчиваются ключевым словом *END*. Между ними могут находиться объявления других объектов, а также значения атрибутов объекта. Значение атрибута объекта начинается с имени атрибута, после которого через разделитель идет значение атрибута. Определение значения атрибута заканчивается символом перевода строки. Все названия объектов, атрибутов и ключевое слово *END* должны быть в верхнем регистре.

Для работы с файлами описания карты создан программный интерфейс, который представляет объекты карты, определенные в файле описания карты в виде экземпляров классов языка программирования высокого уровня Perl. Этот интерфейс позволяет добавлять и удалять из файла описания карты новые объекты, модифицировать атрибуты уже существующих объектов, сохранять произведенные изменения в файл. Этот программный интерфейс позволяет динамически модифицировать файлы описания карты, что дает возможность создавать дополнительные приложения для редактирования существующих карт и создания новых используя HTML-браузер пользователя, исключая необходимость изучения структуры файла описания карты.

Ниже приводится описание объектов определенных для файлов описания карты.

MAP – это корневой объект файла описания карты. Он содержит в себе определение всех других возможных объектов. Атрибуты этого объекта описывают свойства для всей карты в целом.

LAYER – этот объект описывает свойства слоя карты такие как классы объектов внутри слоя, включен ли слой по умолчанию, минимальный и максимальный масштаб при котором отображается данный слой карты, фильтр геометрических примитивов на основе атрибутивной информации, источник данных для слоя, тип геометрических примитивов, составляющих слой, а также атрибут, на основании которого будут определяться классы геометрических примитивов для слоя карты.

CLASS – этот объект отвечает за отображение всех или специально выбранных на основе атрибутивной информации геометрических примитивов, принадлежащих определенному слою карты. Каждый слой карты, определенный объектом *LAYER*, может содержать множество объектов *CLASS*. Атрибуты этого объекта определяют критерий для объединения множества геометрических примитивов, составляющих слой карты, в класс, цвета для отображения границы и внутренней области полигонов и линий, способ отображения точечных геометрических примитивов. Кроме того, атрибуты данного объекта позволяют определить максимальный и минимальный масштаб, при котором

отображаются геометрические примитивы, составляющие класс, и шаблоны, используемые для отображения результатов пространственного поиска по геометрическим примитивам, составляющим класс.

LEGEND – этот объект определяет свойства легенды карты. Атрибуты этого объекта позволяют определить размеры ключей легенды, расстояния между ними, цвет фона легенды, прозрачность фона легенды, а также используемый шрифт для отображения надписей. Легенда отображает все классы активных слоев карты. При этом цвета ключей карты соответствуют цветам, указанным в определении соответствующего класса.

MINIMAP – этот объект отвечает за отображение миникарты. Атрибуты этого объекта определяют размеры объемлющего прямоугольника миникарты, цвет линий на ней, прозрачность и размеры миникарты в пикселях, минимальный и максимальный размер рамки, а также символ для отображения текущего центра карты в случае, когда рамка становится слишком маленькой.

SCALEBAR – этот объект отвечает за отображение линейки масштаба. Атрибуты этого объекта позволяют определить цвет фона линейки масштаба, прозрачность фона, размеры линейки, количество интервалов, единицы измерения длин, стиль отображения линейки масштаба, стиль подписей.

LABEL – Этот объект отвечает за отображение надписей. Он используется при создании миникарты и линейки масштаба. Также он позволяет создавать надписи на самой карте. Атрибуты этого объекта позволяют определить цвет фона надписи, цвет текста надписи, прозрачность фона надписи, размер надписи, положение надписи относительно изображения, на котором она отображается, угол, под которым она отображается, используется ли сглаживание лестничного эффекта.

SYMBOL – этот объект отвечает за отображение символов на карте. В файле описания карты может быть объявлено множество символов, ссылка на каждый конкретный символ производится по его имени. Эти объекты используются для отображения точечных слоев, для установки символов на слоях линий, а также для установки символов на миникарте. Атрибуты этого объекта определяют тип, размеры, цвет, прозрачность символа. В качестве символа может быть использовано графическое изображение в формате JPEG или PNG.

METADATA – этот объект служит для хранения произвольных метаданных для объектов *MAP*, *LAYER*, *CLASS*. Атрибутами этого объекта являются значения метаданных.

4.3 Система визуализации хранимой пространственной информации

4.3.1 Общие сведения

Основной задачей системы визуализации пространственной информации является графическое представление пространственной информации, полученной от системы хранения пространственной информации, используя для этого способы представления этой информации указанные в файле описания карты. Система визуализации создает изображения содержащие карту, миникарту, легенду, линейку масштаба, используя описание их вида, указанные в файле описания карты.

Для отображения карты системе визуализации передается последовательности геометрических примитивов принадлежащих классам всех активных (видимых) слоев карты, а также описание вида этих примитивов, указанное в файле описания карты. Описание вида содержит цвет фона, границы примитивов, описание надписи, которую необходимо отобразить на примитиве, описание символа, который необходимо отобразить на примитиве. При отображении примитива векторные координаты примитива переводятся в пиксельные координаты на основании размера охватывающего карту прямоугольника и размеров карты в пикселях. После перевода координат примитив растеризуется в соответствии с указанными цветом фона и границы, затем на него наносится надпись в соответствии с ее описанием, затем наносится символ, в соответствии с его описанием. Если описание надписи или символа опущены, то на примитив они не наносятся. Если примитивом является линия, то надпись автоматически отображается вдоль линии, если не указано иное, символ автоматически помещается по середине линии. Если примитивом является полигон, то символ отображается в центре охватывающего прямоугольника. Последовательности геометрических примитивов и описания их внешнего вида передаются согласно порядку видимости слоев, указанному в файле описания карты.

Для отображения миникарты системе растеризации передается изображение, содержащие фон миникарты. На основании текущего центра карты, масштаба и размеров охватывающего прямоугольника миникарты вычисляется позиция и размер рамки, отображающей видимую часть карты относительно фонового изображения. После вычисления координат и размеров рамки в пикселях она отображается поверх фонового изображения указанным в файле описания карты цветом. Если размер рамки слишком маленький, то позиция текущего центра карты отображается указанным в файле описания карты символом.

Для отображения легенды используется описание внешнего вида классов, видимых слоев, а также описание внешнего вида ключей легенды. Если видимый класс геометрических примитивов состоит из полигонов, то ключ легенды для этого класса представляет собой прямоугольник, цвет фона которого совпадает с цветом определенным для фона класса геометрических примитивов, цвет границы совпадает с цветом, определенным для границы класса геометрических примитивов. Если видимый класс геометрических примитивов состоит из линий, то ключ легенды для этого класса представляет собой прямоугольник, цвет фона которого указан в файле описания карты в описании цвета фона легенды, внутри прямоугольника находится зигзагообразная линия, цвет которой совпадает с цветом, определенным для цвета линий этого класса. Если видимый класс геометрических примитивов состоит из точек, то ключ легенды для этого слоя представляет символ, определенный в файле описания карты для представления примитивов этого класса. Рядом с каждым ключом отображается надпись, формат которой указана в файле описания карты в описании легенды. Для надписи используется свойство *NAME* класса.

Для отображения линейки масштаба системе визуализации передается описание линейки масштаба из файла описания карты, размер карты в пикселях, размер текущего охватывающего прямоугольника видимой карты. На основе этой информации вычисляется размерность одного экранного пикселя в указанных в файле описания карты в описании линейки масштаба единицах измерения. Затем отображается линейка масштаба в соответствии с указанным в файле описания карты в описании линейки масштаба типом линейки, количеством интервалов, цветом линейки, цветом фона, прозрачностью фона. Надписи на линейке масштаба отображаются в соответствии с описанием надписи для линейки масштаба.

Для растеризации геометрических примитивов, создания и поворота надписей, наложения изображений друг на друга, борьбы с лестничным эффектом при повороте надписей, операций кодирования и декодирования форматов изображений, операций ввода вывода используется свободно распространяемая библиотека классов ImageMagick [1], которая обладает богатыми возможностями и полностью удовлетворяет требованиям системы визуализации пространственной информации.

4.3.2 ImageMagick

ImageMagick – это свободно распространяемая библиотека функций для обработки для рисования примитивов, отображения текста, а также для наложения различных видов эффектов на изображение, таких как поворот, переворот, добавление шумов, размытие, изменение яркости и контрастности, изменение цветовой гаммы, наложение изображений и десятков других. Также эта библиотека позволяет импортировать и экспортировать графические изображения в большинство используемых графических форматов. Эта библиотека функций имеет программный интерфейс представленный в виде библиотеки классов для большинства объектно-ориентированных языков программирования, в частности для Perl и C++.

Библиотека классов для C++ называется Magick++. Magick++ разрабатывался специально с целью создания на его основе коммерческих программных продуктов. Magick++ предоставляет простой программный интерфейс на C++ к библиотеке функций ImageMagick, которая поддерживает чтение и запись огромного количества графических форматов, позволяя выполнять широкий спектр традиционных операций обработки графических изображений.

Центральным классом в Magick++ является класс *Image*. Класс *Image* предоставляет методы для манипуляций над отдельным изображением. Для манипуляций с множеством изображений Magick++, а также для работы с графическими форматами, содержащими более одного кадра (анимированный GIF, MPEG) Magick++ предоставляет STL-совместимый (Standard Template Library [3]) набор алгоритмов.

Класс *Image* поддерживает управление памятью со счетчиком ссылок, которая поддерживает семантику встроенных типов данных, таких как *int*. Так как класс *Image* управляет своей памятью самостоятельно, то экземпляры этого класса лучше создавать при помощи автоматического создания экземпляров классов C++ на стеке. Это позволяет большинству программ, использующих Magick++, обходиться без указателей, что упрощает реализацию и позволяет избежать риска, связанного с использованием указателей. В случае использования автоматического выделения памяти под экземпляры класса *Image* на стеке программа становится более безопасной и совместимой с точки зрения возникновения исключительных ситуаций, так и с точки зрения многопоточности.

Класс *Image* использует множество дополнительных классов в качестве аргументов своих методов. Цвет указывается при помощи класса *Color*. Цвета, указанные в виде строки формата X11, неявно конвертируются в экземпляры класса *Color*. Геометрические аргументы, которые определяют ширину, высоту, смещение указываются с помощью


класса *Geometry* и его потомков. Подобно классу *Color*, строки, содержащие геометрические параметры в формате X11, неявно конвертируются в экземпляры класса *Geometry*. Двумерные графические объекты определяются при помощи класса *Drawable* и его потомков.

Обработка ошибок осуществляется при помощи исключений C++ классами, унаследованными от класса *Exception*, который сам унаследован от стандартного класса C++. Обработка исключений в Magick++ позволяет создавать один блок перехвата исключений на множество вызовов методов классов Magick++.

Ниже приводится список базовых классов Magick++ с кратким описанием класса и потомков.

Таблица 3 - Описание основных классов библиотеки классов Magick++

Имя класса	Описание
Blob	Binary Large Object container (контейнер больших двоичных объектов). Этот класс предоставляет средства для хранения любой неструктурированной информации. Он обладает очень простой функциональностью, позволяя сохранить данные с указанного указателя и получить размер текущих сохраненных данных. Но при этом он сам занимается управлением своей памяти, что придает ему такие же полезные свойства класса <i>Image</i> .
CoderInfo	Предоставляет информацию о поддерживаемых библиотекой ImageMagick форматах графических изображений.
Color	Базовый класс для цвета в Magick++. Это простой класс, который содержит значение красного, синего, зеленого, прозрачности. От него наследуются классы <i>ColorRGB</i> , <i>ColorGray</i> , <i>ColorMono</i> , <i>ColorYUV</i> , экземпляры которых используются пользователем.
Drawable	Этот класс предоставляет удобный интерфейс для подготовки векторных, растровых и текстовых примитивов для передачи их методу <i>Image::draw()</i> . Этот класс является базовым виртуальным классом. Все экземпляры его потомков представляют собой один простой объект, который может быть нарисован. Экземпляр потомка этого класса может быть нарисован при помощи одного вызова метода <i>draw</i> класса <i>Image</i> или множество этих объектов может быть нарисовано при помощи передачи методу <i>draw()</i> класса <i>Image</i> списка этих объектов.
Exception	Предоставляет собой базовый класс для классов объектов, создаваемых при возникновении исключительных ситуаций в библиотек ImageMagick.
Geometry	Этот класс предоставляет удобные средства для указания геометрических аргументов. Этот объект может быть инициализирован неявным конвертированием строки формата X11 в экземпляр класса <i>Geometry</i> .
Image	Этот класс основной класс в Magick++ и представляет собой одно изображение. Множество манипуляций может быть проделано над экземпляром класса <i>Image</i> . Также класс <i>Image</i> содержит атрибуты, влияющие на растеризацию изображения. Этот класс также предоставляет низкоуровневый доступ к изображению на уровне пикселей при помощи класса <i>Pixels</i> .
Montage	Монтаж – это графическое изображение, объединяющее в себе множество уменьшенных копий других изображений с подписями. Класс <i>Montage</i>

	<p>предоставляет удобный интерфейс для создания подобных изображений.</p>  <p style="text-align: center;">Рисунок 4.2 - Пример монтажа</p>
<p>Pixels</p>	<p>Этот класс предоставляет удобный доступ напрямую к пикселям изображения.</p>
<p>STL</p>	<p>Magick++ предоставляет набор STL-совместимых алгоритмов для обработки множества изображений, хранимых в STL-контейнерах. Также предоставляются набор унарных функций-алгоритмов для применения к каждому отдельному изображению, хранимому в STL-контейнере, алгоритмами обработки контейнера.</p>
<p>TypeMetric</p>	<p>Предоставляет средства для работы с атрибутами используемых шрифтов такими как высота, ширина символа, отступы перед и после символа и прочее.</p>

4.4 Система организации интерфейса пользователя

4.4.1 Общая информация

Основной задачей этой системы является вывод созданных системой визуализации пространственной информации изображений пользователю, предоставление средств для управления отображаемой информацией, предоставления средств для редактирования хранимой пространственной информации и метаданных, а также обработка пространственных запросов пользователя.

Так как в качестве средства отображения интерфейса является HTML-браузер клиента, то все элементы интерфейса созданы при помощи языка HTML. Для отправки запросов пользователя приложению используется стандартный для HTML механизм передачи данных при помощи кодирования передаваемых данных в строку содержащие пары, состоящие из имени параметра и его значения, и отправки этой строки в теле HTML запроса типа POST к серверу. Для кодирования данных, передаваемых серверу, в основном используются средства HTML-браузера, такие как автоматическое кодирование данных содержащихся в формах HTML при посылке из содержимого на web-сервер.

В качестве технологии взаимодействия клиента и сервера согласно техническим требованиям к проекту используется CGI. Для упрощения работы с этой технологией и избежания трудностей, связанных с разработкой пользовательского интерфейса с приложением, которые возникают при использовании этой технологии, используется библиотека классов CGI Core, разработанная фирмой Интис специально для этого проекта. Эта библиотека классов постоянно развивается и используется фирмой Интис для создания коммерческих программных продуктов, созданных на основе технологии взаимодействия клиента и сервера CGI.

Интерфейс пользователя состоит из двух основных частей. Первая часть доступна всем пользователям и позволяет выбирать карты для просмотра, выбирать отображаемые слои, перемещаться по карте, масштабировать карту, изменять настройки цветов для отображения карты, а также создавать пространственные запросы. Вторая часть доступна только привилегированным пользователям и предоставляет средства для администрирования хранимой пространственной информации и метаданных. Эта часть позволяет добавлять и удалять новые карты, добавлять, изменять атрибуты и удалять слои существующих карт, а также модифицировать атрибутивную информацию для каждого слоя. Здесь также можно создавать классы геометрических примитивов, принадлежащих одному слою карты, на основе атрибутивной информации этого слоя.

4.5 Разработка CGI приложений

Интернет ГИС портал реализован как CGI приложение: все программы исполняются на сервере, взаимодействие с пользователем реализуется через CGI интерфейс, а диалог с пользователем представлен в виде набора гипертекстовых документов. В данном разделе рассмотрены CGI интерфейс, специфика CGI программирования и обзор библиотеки классов CGI Core, с помощью которой ГИС портал - задача данного класса, была успешно реализована.

4.5.1 Обзор CGI интерфейса

Большая часть общения пользователя с WWW сводится к тому, что веб-браузер передает веб-серверу URI (Universal Resource Identifier) очередной веб-страницы, сервер находит ее в сети и передает браузеру, который ее отображает. Сами веб-страницы реализуются на языке разметки гипертекста HTML (Hyper Text Markup Language), который, позволяет создавать только статичные страницы. Часто этого достаточно, но существует широкий круг задач, которые не может быть решены с использованием одного HTML, когда необходимо провести некоторые вычисления, например запрос к базе данных с целью выборки данных, отвечающих заданному критерию, указанному пользователем. На сегодняшний день существуют различные подходы для решения таких задач, наиболее распространенным является следующий:

1. Передача серверу запроса, по которому он запустит некоторое приложение.
2. Передача параметров запроса запущенному приложению.
3. Передача результатов работы приложения обратно к клиенту.

Интерфейс CGI обеспечивает такое клиент-сервер взаимодействие. Общая схема взаимодействия клиента с сервером проиллюстрирована на рисунке:

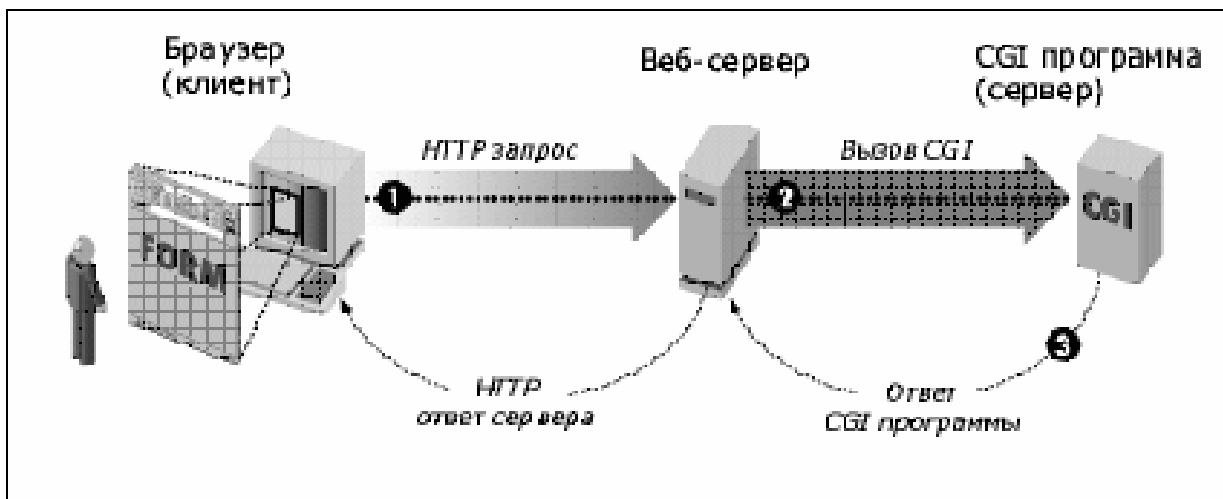


Рисунок 4.3 - Интерфейс CGI

4.5.2 Проектирование и разработка CGI приложений

CGI приложение может быть реализовано на любом языке программирования, главное, чтобы бы веб-сервер мог его запустить. Параметры запроса передаются приложению на вход, через переменные окружения и стандартный вход (STDIN), результат работы программы ожидается веб-сервером на ее стандартном выходе (STDOUT). Таким образом, может показаться, что разработка CGI приложения существенно не отличается от разработки обычного приложения: пользователь последовательно указывает параметры входа, программы обрабатывает вход и выдает

результат работы, пользователь продолжает с ней работу или завершает ее. Такой сеанс работы называется сессией, в ходе сессии программа в каждый момент времени находится в некотором состоянии. К сожалению, сеанс работы CGI приложения ограничен, он начинается тогда, когда сервер запускает его и передает ему параметры входа, и завершается тогда, когда вход обработан, а сервер получил выход. Таким образом, CGI приложение теряет свое состояние. Впрочем, этого достаточно, если взаимодействие с пользователем ограничено простым «запрос-ответ», и в каждый момент времени не зависит от того, что было раньше.

Спецификой разработки CGI приложения, обеспечивающего сложный диалог с пользователем, является необходимость поддержки сессий. Сессия содержит в себе текущее состояние программы, необходимые переменные, имя пользователя, уровень доступа и так далее. Сессия обычно хранится на жестком диске сервера и идентифицируется уникальным номером. Указав номер сессии во входном параметре к CGI приложению, можно загрузить предыдущее состояние программы и передать ему новые параметры на вход, с помощью такой схемы можно «поддерживать» разговор клиента и сервера. Заметим, что номер сессии должен быть уникальным, тогда с одной и той же программой смогут работать несколько пользователей одновременно, при этом каждый будет указывать номер сессии, который выдала ему CGI программа. Необходимо предусмотреть методы добавления параметров в сессию и их удаления. Так же у каждой сессии должен быть параметр «время последнего использования», используя который можно удалять файлы сессий, неиспользуемые в течение долгого времени.

Вторым важным моментом является необходимость разработки темплейтной системы. Так как на выходе CGI программы обычно должен быть HTML документ, в котором некоторые его части изменяются программой, делая документ динамическим. Такой документ можно хранить в самой программе и при необходимости выдавать его на выход, автоматически вставляя в изменяемые части значения вычисленных переменных. Однако такое перемешивание программного кода и HTML кода становится неудобным, если требуется изменять дизайн HTML документа. Эту работу обычно выполняет дизайнер или HTML кодер, которые не знают кода, на котором написана программа, тогда приходится привлекать к работе программиста, в итоге время и стоимость изменений значительно увеличивается. Решением является хранение отдельное хранение HTML документов и программного кода. В изменяемые части HTML вставляются особые строки, которые анализируются парсером темплейтной системы и заменяются значениями вычисляемых переменных.

Другим важным моментом является задача обеспечения безопасности, например коммерческие CGI приложения часто обращаются к базе данных клиентов для обработки номеров кредитных карт, личных данных о клиентах. Злоумышленник, сформировав некорректный запрос к базе данных, может получить доступ к таким конфиденциальным данным или удалить их. Не корректно написанный CGI скрипт может позволить злоумышленнику получить доступ к любым данным на сервере, например к базе данных безопасности операционной системы, зациклить программу и этим вызвать перегрузку сервера. При написании CGI программы необходимо чрезвычайно внимательно отнестись к вопросу безопасности.

4.5.3 Обзор CGI Core

На основании вышесказанного можно понять, что разработка даже сравнительно простого CGI приложения может потребовать много времени, конечно, если преследуется цель написания качественного, масштабируемого и многоплатформенного кода. Для решения вышеперечисленных задач в данной работе и последующих, была разработана библиотека классов CGI Core, на основе которой разрабатывались все CGI приложения интернет ГИС портала.

Библиотека реализована на языке Perl, и использует множество модулей с сервера <http://www.CPAN.org/>, доступных для свободного использования. Использование объектно-ориентированного подхода (ООП) к разработке позволило сохранить библиотеку «открытой» для последующих расширений и модернизаций, что позволяет легко подключать модули, разработанные сторонними разработчиками и расширять функциональность программ. Библиотека поддерживает динамическую загрузку модулей «по требованию» для экономии оперативной памяти, и на данный момент обеспечивает следующую функциональность:

1. поддержка сессий на базе CGI::Persistent [12]
2. модуль безопасности, работающий с базами данных безопасности на базе СУБД MySQL [13] и AsciiDB [14]
3. темлейтная система на базе HTML::Template [15]
4. конфигурирование системы через Config::IniFiles [16]
5. поддержка СУБД MySQL и AsciiDB
6. модуль для проверки значений полей формы
7. многоязычность, автоматическое переключение HTML шаблонов в зависимости от языков, поддерживаемых браузером пользователя, функции для локализации: отображение даты и времени на разных языках
8. многоплатформенная почтовая подсистема на базе Mail::Mailer [17], обеспечивающая отправку электронной почты с помощью следующих почтовых агентов: sendmail, mail или по протоколу SMTP
9. модули для платформонезависимой работы с файловой системой: IO::Dir [18], IO::LockableFile, File::Spec
10. модули WDDX [19], обеспечивающие экспорт структур данных Perl в структуры данных других языков: XML, JavaScript, .
11. загрузка файлов через HTTP (upload, download)

Класс Core унаследован от класса CGI::Application [20], философия которого отталкивается от идеи, что любое CGI приложение можно организовать как определенный набор «режимов исполнения» (Run-modes). Каждый режим исполнения – это грубый аналог одного экрана (форма, результат выполнения программы и так далее). Все режимы исполнения управляются одним «модулем приложения» (Application Module), который является Perl модулем, который веб-сервер запускает как CGI приложение. Данная методология является инверсией «вложенной» (Embedded) философии (ASP, JSP, EmbPerl, Mason и другие), в которой есть «страницы» для каждого состояния приложения, которые «управляют» программой. CGI::Application модуль приложения управляет страницами и код всего приложения находится в одном месте, а не разбросан по множеству файлов.

Разработка CGI Core была начата в августе 2002 года, в работе приняли участие: Милованцев П.Б., Антипин М.Г., Малахов Р.Ю., Чухломин А.А. CGI Core постоянно развивается и на его основе помимо ГИС портала разработаны: интернет чат, система онлайн продаж электронных продуктов, множество небольших скриптов.

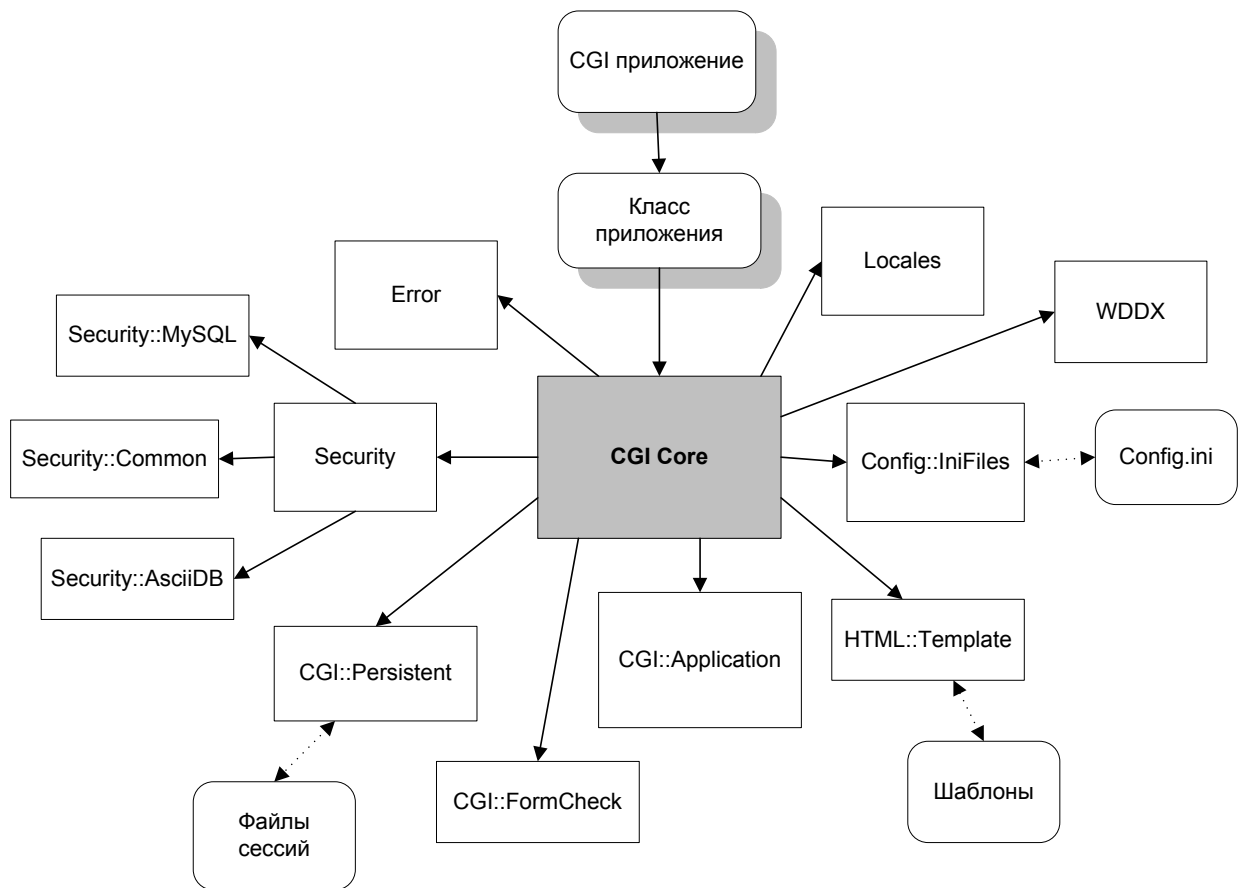


Рисунок 4.4 - Схема CGI приложения на основе CGI Core

4.6 Архитектура приложения

Приложение состоит из трех основных частей, выполняющих разные по степени ресурсоемкости задачи.

Система визуализации пространственной информации занимается динамическим созданием графических изображений, возможно обладающих большими размерами. Эта задача наиболее требовательна к ресурсам сервера, на котором она установлена, в частности к производительности центрального процессора и системы оперативной памяти. Также эта система получает больше всего запросов, так как на каждый запрос пользователя необходимо создать заново все графические изображения, связанные с отображением карты, и длительная работа этой системы приведет к увеличению времени ожидания клиентом ответа сервера на запросы. Таким образом, это наиболее ресурсоемкая часть приложения и требования производительности являются для нее основными.

Система хранения пространственной информации и метаданных занимается в основном операциями ввода-вывода, используя для этого в основном файловую систему сервера. При этом система хранения пространственной информации занимается также пространственным и атрибутивным поиском по хранимой информации, а также интерпретацией файлов описания карт. Эти задачи также являются достаточно ресурсоемкими, и могут существенно повлиять на время получения пользователем ответа на запрос.

С другой стороны, пользователь большую часть времени занимается изучением полученных от сервера графических изображений и манипуляциями с интерфейсом приложения. Запросы на сервер пользователь посылает достаточно редко, но хочет быстро получать от сервера ответы на них. Поэтому система организации пользовательского интерфейса получает малое количество запросов от пользователя. При этом обработка этой системой запроса пользователя перед передачей его системам хранения пространственной информации и метаданных и системе визуализации не является ресурсоемкой задачей. Поэтому требования к производительности системы организации интерфейса пользователя не являются существенными.

В связи с вышеизложенным было принято решение создать следующую архитектуру приложения. Функции систем, требования к производительности которых являются существенными, вынести в написанную на C++ динамически загружаемую библиотеку функций *mapview.so*. Эта библиотека предоставляет функции для системы визуализации пространственной информации, такие как растеризация геометрических примитивов, растеризация текста, наложения изображений друг на друга и прочие, функции для работы системы хранения пространственной информации, такие как чтение и запись ESRI Shapefile, чтение, интерпретация и запись файлов описания карты, пространственного и атрибутивного поиска хранимой пространственной информации. Решение использовать C++ а не Perl для создания этой библиотеки объясняется тем, что Perl – язык интерпретируемый, и скорость исполнения приложения на языке Perl порой в сотни раз ниже скорости исполнения аналогичных приложения скомпилированных под программно-аппаратную платформу, на которой они исполняются. А так как в библиотеке *mapview.so* собраны функции, требования к производительности которых наиболее существенны, то необходимо было использовать для создания этой библиотеки наиболее производительные программные средства.

Для создания интерфейса пользователя было принято решение использовать интерпретируемый язык Perl, в связи с его высочайшим удобством для программиста, огромной библиотекой CPAN (Comprehensive Perl Archive Network) [6] свободно распространяемых модулей для этого языка, позволяющих решать большинство задач

связанных с CGI программированием и программированием вообще, наличием встроенного в язык менеджер памяти со сборщиком мусора, наличием встроенных средств для работы с регулярными выражениями, простотой создания интерфейса с динамически загружаемыми библиотеками. Для библиотеки *mapview.so* был создан объектно-ориентированный интерфейс, представляющий функции библиотеки *mapview.so* в виде методов объектов основных используемых в приложении классов объектов. Создание объектно-ориентированного программного интерфейса ко всем функциями динамически загружаемой библиотекой *mapview.so* облегчает создание системы организации интерфейса пользователя и создает средства для изменения хранимой пространственной информации и метаданных и механизмов создания изображения средствами интерфейса пользователя.

Благодаря такому подходу существенно упрощается создание интерфейса пользователя благодаря использованию средств Perl, CGI Core и объектно-ориентированного программного интерфейса с библиотекой *mapview.so*, облегчается поддержка приложения и дальнейшее расширение функциональности интерфейса пользователя приложения. В то же время удовлетворяются требования производительности, т.к. на интерпретируемом языке Perl работает только система первичной обработки запросов клиента и интерфейс вызовов функций из динамически загружаемой библиотеки *mapview.so*.

Для создания объектно-ориентированного интерфейса с динамически загружаемой библиотекой функций *mapview.so* был использован инструмент разработки SWIG (Simplified Wrapper and Interface Generator) [4]. SWIG – это компилятор интерфейсов, который соединяет программы написанные на С или С++ с интерпретируемыми языками программирования, такими как Perl, Python, Ruby, Tcl, C#. Для своей работы SWIG использует объявления функций из заголовочных файлов С или С++, и использует их для генерации специфичного для каждого интерпретируемого языка кода, необходимого для доступа к функциям, написанным на С или С++. Также этот инструмент содержит множество средств настройки процесса создания интерфейса интерпретируемого языка программирования с кодом, написанным на С или С++.

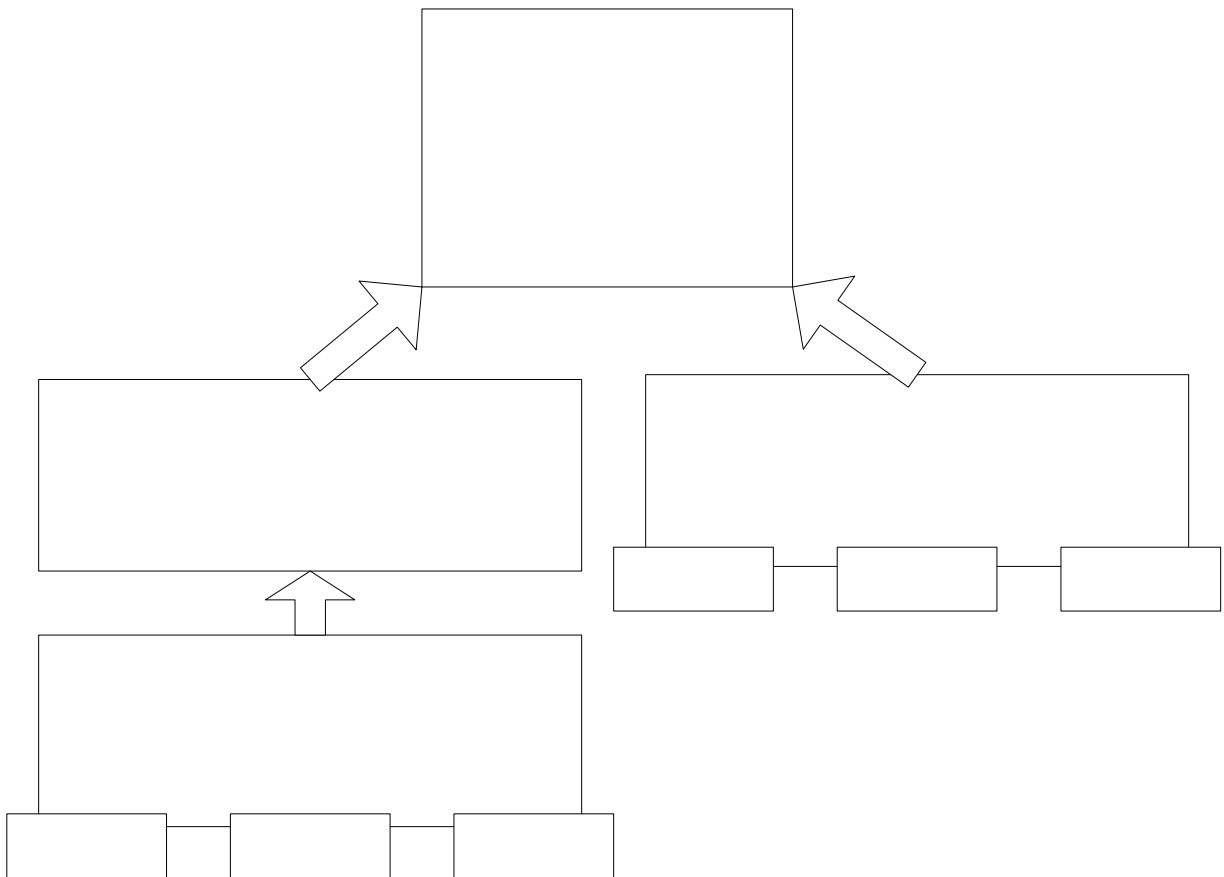


Рисунок 4.5 – Архитектура приложения

- **MapView.pm** – модуль Perl который содержит описание объектно-ориентированного интерфейса с функциями динамически загружаемой библиотеки *mapview.so*.
- **XBase.so** – свободно распространяемая библиотека для работы с базами данных формата dBase. Предоставляет средства для чтения и записи, средства навигационного и спецификационного поиска.
- **Proj4.so** – свободно распространяемая библиотека для работы с различными топологическими проекциями.
- **Magick++** - библиотека классов на C++ предоставляющих объектно-ориентированный интерфейс к функциям графической библиотеки ImageMagick.
- **MapView.so** – библиотека функций, используемых для растеризации изображений, работы с файлами описания карты, с ESRI Shapefile.
- **CGI Core** – библиотека классов, используемая для создания CGI приложений.
- **CGI::Application** – базовый класс CGI Core.
- **CGI::Persistent** – класс Perl для создания CGI приложений с сохранением состояний. Используется для работы с сессией пользователя приложения.
- **HTML::Template** – класс Perl, используемый для работы с шаблонами и интерфейсы создаваемого при помощи HTML.

Mapvi

Заключение

В результате проделанной работы были изучены аппаратные средства для получения атрибутивных данных имеющихся в распоряжении Института Оптического Мониторинга СО РАН, проведен анализ существующих в разрозненном виде баз данных Института Оптического Мониторинга. На основании анализа и изучения аппаратных средств составлен перечень технических требований к программному комплексу, определена его будущая архитектура, основные компоненты и система их взаимодействия.

Компонента для визуализации пространственно-распределенной информации была реализована в виде библиотеки классов языка Perl, которая является объектно-ориентированным программным интерфейсом к библиотеке функций, написанной на C++ с использованием свободно распространяемых библиотек функций. Такая реализация компоненты визуализации позволяет отделить ее основную функциональность от интерфейса пользователя, использовать функциональность этой библиотеки классов в различных областях, связанных с работой с файлами формата ESRI Shapefile, отображением геометрических примитивов. Так как библиотека классов предоставляет средства для работы со всеми этапами процесса визуализации начиная от чтения-записи пространственно-распределенной информации из файлов формата ESRI Shapefile, чтения-записи метаданных и заканчивая растеризацией карты, легенды, линейки масштаба и миникарты, то на основе этой библиотеки классов можно создавать богатые по функциональности приложения, позволяющие не только отображать пространственно-распределенную информацию, но и полноценно редактировать пространственно-распределенную информацию и связанные с ней метаданные. Благодаря этим свойствам после незначительной доработки и документирования, эту библиотек классов можно использовать как коммерческий продукт, используемый для отображения и редактирования пространственно-распределенной информации как в виде CGI приложения, так и в других видах приложений.

На основе вышеописанной библиотеки классов было создано CGI приложение, позволяющее пользователю просматривать пространственно-распределенную информацию в виде карты, создавать пространственные запросы для просмотра атрибутивной информации, а также модифицировать метаданные, связанные с отображаемой пространственной информацией. Для создания этого приложения была создана библиотека классов языка Perl, в значительной мере облегчающая разработку CGI приложения, позволяющая абстрагироваться от множества трудностей, связанных с разработкой CGI приложений, создавать более надежные, масштабируемые, функционально богатые, поддерживаемые CGI приложения. CGI Core используется во многих коммерческих программных продуктах, созданных ООО «Интис». Также планируется дальнейшее коммерческое использование самой библиотеки классов CGI Core.

В результате работы был получен большой опыт разработки коммерческих программных продуктов для операционной системы Linux, объектно-ориентированного анализа и программирования, разработки CGI приложений.

Список использованных источников

1. ImageMagick – Convert, Edit and Compose Images
<http://www.imagemagick.org>
2. ESRI Shapefile technical format description
<http://www.esri.com/library/whitepapers/pdfs/shapefile.pdf>
3. Standard Template Library Programmer’s Guide
<http://www.sgi.com/tech/stl/>
4. Wrapper and Interface Generator
<http://www.swig.org>
5. ESRI Corporation
<http://www.esri.com>
6. Comprehensive Perl Archive Network
<http://www.cpan.org>
7. MapInfo Corporation
<http://www.mapinfo.com>
8. Microsoft Data Access Technologies
<http://www.microsoft.com/data>
9. Active Server Pages Development
<http://www.asp101.com>
10. Perl Mongers
<http://www.perl.org>
11. Институт Оптики Атмосферы СО РАН. Грант Intas
<http://www.iao.ru/?dc=structure/dwp/llde&fc=atmos&dm=structure/dwp/llde&fm=menu>
12. CGI::Persistent -- Transparent state persistence for CGI applications
<http://search.cpan.org/author/VIPUL/CGI-Persistent-0.23/>
13. MySQL: The World's Most Popular Open Source Database
<http://www.mysql.com/>
14. Jose A. Rodriguez / AsciiDB-TagFile-1.06
<http://search.cpan.org/author/JOSERODR/AsciiDB-TagFile-1.06/>
15. Sam Tregar / HTML-Template-2.6
<http://search.cpan.org/author/SAMTREGAR/HTML-Template-2.6/>
16. Jeremy Wadsack / Config-IniFiles-2.38
<http://search.cpan.org/author/WADG/Config-IniFiles-2.38/>
17. Mark Overmeer / MailTools-1.58
<http://search.cpan.org/author/MARKOV/MailTools-1.58/>
18. Graham Barr / IO-1.20

<http://search.cpan.org/author/GBARR/IO-1.20/>

19. Scott Guelich / WDDX-1.01

<http://search.cpan.org/author/GUELICH/WDDX-1.01/>

20. Jesse Erlbaum / CGI-Application-3.0

<http://search.cpan.org/author/JERLBAUM/CGI-Application-3.0/>

21. THE 7th HITRAN DATABASE CONFERENCE

<http://cfa-www.harvard.edu/hitran/Proceedings-seventh.pdf>

22. HITRAN

<http://www.hitran.com/>

23. Холзнер. С. Perl: специальный справочник – СПб: Питер, 2001. – 496с.:ил.

24. Гулич С., Гундавара Ш., Бирзнекс Г. CGI программирование на Perl. – СПб: Символ-Плюс, 2001. 480с., ил.

25. Г.Буч Объектно-ориентированный анализ и проектирование с примерами приложений на C++, 2-е изд. – М.: «Издательство Бином», 2000. – 560с., ил.

26. Роджерс Д., Адамс. Дж. Математические основы машинной графики: М.: Мир, 2001. – 604с., ил.

Приложение А. Руководство пользователя

А.1 Пользовательская часть

Для использования программы необходима рабочая станция с любым каналом доступа в Интернет и установленным на ней программным обеспечением для работы с WEB, в частности, необходимо наличие графического веб-браузера.

Для начала работы необходимо открыть при помощи графического браузера, установленного на рабочей станции, ресурс с URL <http://arena.intis.tomsk.ru/max/gisportal>. В появившемся окне вам будет предложено выбрать одну из имеющихся на сервере географических карт регионов. После того, как желаемая карта была выбрана, можно нажать кнопку «Продолжить». После этого браузер откроет основное рабочее окно.

Основное рабочее окно содержит две основных панели. В левой панели отображается графическое изображение карты региона. Заголовок этой панели отображает название карты региона. Графическое изображение карты региона может быть использовано для перемещения по карте, а также для масштабирования. Для этого надо щелкнуть мышкой в желаемой точке изображения. Указанная точка становится новым центром перемещенного или масштабированного изображения. Для определения поведения приложения при щелчке мышкой на графическое изображение карты региона используется панель инструментов. Под графическим изображением карты региона находится линейка масштаба.

Панель инструментов содержит несколько именованных разделов. Раздел «Слои» содержит список доступных для отображения слоев карты. Перед названием каждого слоя стоит «CheckBox», который отображает, выбран ли данный слой, и позволяет выбирать слой, если он не выбран. Для отображения слоев, после внесения изменений, необходимо нажать кнопку «Применить» внизу этого раздела панели инструментов.

Раздел «Режим» содержит переключатель двух режимов работы. При включенном режиме «Просмотр» при щелчке мыши на изображении будет происходить масштабирование либо перемещение центра карты. При включенном режиме «Опрос» при щелчке мышки на карте будет происходить пространственный поиск геометрических примитивов в окрестности этой точки. В случае успешного поиска выводится страница с атрибутивной информацией геометрических примитивов, находящихся в окрестности указанной точки.

Следующим разделом панели инструментов является раздел «Масштабирование». Этот раздел используется для изменения масштаба карты, как с использованием графического изображения карты, так и без него. Для этого раздел содержит 3 «RadioButton», которые определяют направление масштабирования, которое будет использоваться при щелчке на графическом изображении карты региона или при нажатии кнопки «Применить» внизу этого раздела панели инструментов.

Последним разделом панели инструментов является раздел «Легенда», в котором отображается легенда для текущих выбранных слоев карты.

А.2 Административная часть

Для использования программы необходима рабочая станция с любым каналом доступа в Интернет и установленным на ней программным обеспечением для работы с WEB, в частности, необходимо наличие графического веб-браузера.

Для начала работы с административной частью приложения необходимо открыть в браузере ресурс с URL http://arena.intis.tomsk.ru/cgi-bin/max/gisportal/admin_viewer.pl. В

появившемся окне будет показан список имеющихся в наличии карт. Для каждой карты отображается имя файла описания карты, Название карты, параметры объемлющего прямоугольника карты и единицы измерения, в которых указываются пространственные координаты геометрических примитивов, составляющих слои карты. Название карты можно изменить, введя новое название и нажав кнопку «сохранить». Для каждой карты также имеется ссылка на окно для работы со слоями карты.

В окне для работы со слоями карты отображаются все определенные в файле описания слои карты и их параметры. Для каждого слоя карты отображается его имя, название, тип геометрических примитивов, составляющих слой, отображается ли слой по умолчанию. Пользователь может изменить название слоя и отображение слоя по умолчанию. Для этого необходимо ввести новое название слоя и выбрать отображения слоя по умолчанию и нажать кнопку «Сохранить». Для каждого слоя имеются ссылки на окно для редактирования атрибутивной информации слоя и на окно для работы с классами слоя.

В окне редактирования атрибутивной информации геометрических примитивов слоя отображается таблица, содержащая все строки таблицы атрибутивной информации. Каждая строка таблицы представляет атрибутивную информацию для одного геометрического примитива слоя. Столбец «ID» соответствует порядковому номеру полигона в файле формата ESRI Shapefile. Для любой строки таблицы можно изменить значения столбцов. Для этого надо ввести их новые значения и нажать кнопку «Сохранить». Вверху страницы имеется ссылка, при переходе на которую все данные, содержащиеся в таблице, конвертируются из кодировки windows-1251 в KOI-8. Это облегчает импорт файлов формата ESRI Shapefile созданных на операционных средах Windows 98/2000/XP.

Окно для работы с классами слоя содержит вверху список классов для выбранного слоя, определенный в файле описания карты. Для каждого класса отображается название класса, цвет, которым отображаются геометрические примитивы, принадлежащие к данному классу, а также строка, которая определяет принадлежность геометрического примитива к данному классу. Цвет отображается в виде трех компонент формата цвета RGB. Пользователь может изменить название класса, строку принадлежности и цвет, которым отображаются геометрические примитивы, принадлежащие классу, изменив соответствующие значения и нажав кнопку «Сохранить».

Под списком классов находится форма для создания нового класса. Все поля в ней обязательные. После заполнения всех полей необходимо нажать кнопку «Добавить». Добавленный класс можно удалить, нажав кнопку «Удалить». Под формой добавления нового класса находится поле, содержащее название столбца, который используется для определения принадлежности к классу. Пользователь может указать любой столбец из таблицы атрибутов для данного слоя, используя список имен столбцов. После выбора желаемого столбца необходимо нажать кнопку «Сохранить».

Под полем выбора столбца находятся формы для добавления классов на основе атрибутивной информации. Список этих форм формируется на основе всех различных значений в столбце, указанном выше. Из этого списка удаляются значения, которые уже присутствуют в строке принадлежности классов, объявленных в файле описания карты. Если полученный список пустой, то формы для создания новых классов не создаются. В форме добавления нового класса можно ввести цвет, которым будут отображаться геометрические примитивы этого класса. Для добавления нового класса необходимо нажать кнопку «Добавить». Добавленный класс в качестве своего названия использует значение столбца, это же значение автоматически устанавливается для строки принадлежности. Добавленный класс появляется в списке классов слоя. Удалить

добавленный класс можно в верхнем списке классов. Для этого для каждого класса из списка объявленных в файле описания карты классов имеется кнопка «Удалить».

Приложение Б. Формат файла описания карты

Файл описания карты представляет собой текстовый файл, имеющий расширение *map*. Формат файла объявления регистрозависимый, имена всех объектов и названия всех атрибутов объектов должны быть указаны в верхнем регистре, за исключением атрибутов объекта *METADATA*. В качестве разделительных символов используются символы пробела и табуляции. Каждая строка файла может содержать комментарий, объявление начала нового объекта, объявление атрибута объекта, ключевое слово *END*. Комментарий это строка, начинающаяся с символа '#'. Объявление начала нового объекта представляет собой строку, содержащую за исключением разделительных символов имя объекта. Объявление атрибута объекта представляет собой строку, начинающуюся с нуля или более разделительных символов, за которыми следует название атрибута объекта, после которого через один или несколько разделительных символов следует значение атрибута. Для предотвращения неоднозначного толкования значения атрибутов можно заключать в двойные кавычки. В этом случае символам двойных кавычек в значении атрибута должен предшествовать символ '\

В файле описания карты могут быть определены нижеописанные объекты.

MAP – это корневой объект файла описания карты. Он содержит в себе определение всех других возможных объектов. Атрибуты этого объекта описывают свойства для всей карты в целом.

Таблица 4 - Атрибуты объекта MAP

Атрибут	Описание
EXTENT [minx] [miny] [maxx] [maxy]	Размеры объемлющего прямоугольника карты. Тип значения – double. Этот атрибут может быть опущен. В этом случае его значение определяется на основании объемлющих прямоугольников всех объявленных в файле описания карты слоев. Этот атрибут используется в том случае, если необходимо показать только часть доступной территории.
IMAGECOLOR [r] [g] [b]	Цвет фона карты. Тип значения – целое от 0 до 255. Задаёт цвет фона карты в формате RGB.
IMAGETYPE [gif png]	Формат создаваемых системой визуализации пространственной информации изображений. Тип значения – строка. Указывает системе визуализации пространственной информации, какой графический формат использовать для создаваемых ей изображений. Этот формат распространяется на все создаваемые изображения.
NAME	Префикс, используемый для имен графических изображений при сохранении их в директорию, доступную веб-серверу. Тип значения – строка.
DATAPATH	Путь до директории содержащий файлы формата ESRI Shapefile. Тип значения – строка. Используется для указания абсолютного пути к директории, содержащей пространственную информацию формата ESRI Shapefile.
SIZE [x] [y]	Размер карты в пикселях. Тип значения – целое без знака.
TRANSPARENT	Включает или выключает прозрачность фона карты. Тип

[on off]	значения – строка.
UNITS [kilometers meters dd]	Указывает систему измерений, в которой заданы векторные координаты геометрических примитивов. Тип значения – строка. Используется в связи с тем, что формат ESRI Shapefile не содержит единиц измерения, в которых заданы координаты геометрических примитивов.

LAYER – этот объект описывает свойства слоя карты такие как классы объектов внутри слоя, включен ли слой по умолчанию, минимальный и максимальный масштаб при котором отображается данный слой карты, фильтр геометрических примитивов на основе атрибутивной информации, источник данных для слоя, тип геометрических примитивов, составляющих слой, а также атрибут, на основании которого будут определяться классы геометрических примитивов для слоя карты. Этот объект может быть объявлен внутри объекта *MAP*.

Таблица 5 - Атрибуты объекта **LAYER**

Атрибут	Описание
CLASSFIELD [string]	Название столбца из связанного со слоем файла формата ESRI Shapefile. Тип значения – строка. Используется совместно с атрибутом <i>FILTER</i> объекта <i>CLASS</i> для определения классов геометрических примитивов внутри одного слоя карты на основании атрибутивной информации примитивов слоя.
DATA [filename]	Абсолютный путь к файлу формата ESRI Shapefile, геометрические примитивы и атрибутивная информация которого составляют слой. Тип значения – строка. Расширение файла может быть опущено.
FOOTER [filename]	Шаблон, используемый в качестве нижней части при создании страницы с результатами пространственного поиска по элементам слоя. Тип значения – строка.
HEADER [filename]	Шаблон, используемый в качестве заголовка при создании страницы с результатами пространственного поиска по элементам слоя. Тип значения – строка.
TEMPLATE [filename]	Шаблон, используемый в качестве основной части при создании страницы с результатами пространственного поиска по элементам слоя. Тип значения – строка. Используется в качестве задания глобальной альтернативы свойству <i>TEMPLATE</i> объекта <i>CLASS</i> .
MAXSCALE [double]	Максимальный масштаб, при котором отображаются элементы данного слоя карты. Тип значения – double. Используется в качестве задания глобальной альтернативы свойству <i>MAXSCALE</i> объекта <i>CLASS</i> .
MINSCALE [double]	Минимальный масштаб, при котором отображаются элементы данного слоя карты. Тип значения – double. Используется в качестве задания глобальной альтернативы свойству <i>MINSCALE</i> объекта <i>CLASS</i> .
NAME [string]	Имя слоя. Тип значения – строка. Используется в качестве уникального идентификатора слоя.
STATUS	Определяет отображается ли слой на карте. Тип значения – строка.

[on off default]	default – слой отображается всегда.
TYPE [point line polygon]	Тип геометрических примитивов, составляющих слой. Тип значения – строка. Может быть опущен.

CLASS – этот объект отвечает за отображение всех или специально выбранных на основе атрибутивной информации геометрических примитивов, принадлежащих определенному слою карты. Каждый слой карты, определенный объектом *LAYER*, может содержать множество объектов *CLASS*. Атрибуты этого объекта определяют критерий для объединения множества геометрических примитивов, составляющих слой карты, в класс, цвета для отображения границы и внутренней области полигонов и линий, способ отображения точечных геометрических примитивов. Кроме того, атрибуты данного объекта позволяют определить максимальный и минимальный масштаб, при котором отображаются геометрические примитивы, составляющие класс, и шаблоны, используемые для отображения результатов пространственного поиска по геометрическим примитивам, составляющим класс. Этот объект может быть объявлен внутри объекта *LAYER*.

Таблица 6 - Атрибуты объекта CLASS

Атрибут	Описание
BACKGROUNDCOLOR [r] [g] [b]	Определяет цвет фона в формате RGB для геометрических примитивов типа polygon. Тип значения целое от 0 до 255.
COLOR [r] [g] [b]	Задаёт цвет границы для геометрических примитивов типа polygon и цвет линий для геометрических примитивов типа line в формате RGB. Тип значения целое от 0 до 255.
FILTER [string]	Определяет класс геометрических примитивов среди всех геометрических примитивов слоя на основании атрибутивной информации. Тип значения – строка. Этот атрибут используется совместно с атрибутом CLASSFIELD объекта LAYER, которому принадлежит класс. Все геометрические примитивы, у которых значения атрибутивной информации для указанного атрибута CLASSFIELD объекта LAYER совпадают со значением атрибута FILTER объекта CLASS, относятся к одному классу.
MAXSCALE [double]	Максимальное значение масштаба, при котором отображаются элементы класса. Тип значения – double. Этот атрибут может быть использован для создания нескольких способов отображения элементов одного слоя при разных масштабах.
MINSCALE [double]	Минимальное значение масштаба, при котором отображаются элементы класса. Тип значения – double. Этот атрибут может быть использован для создания нескольких способов отображения элементов одного слоя при разных масштабах.
SYMBOL [string]	Имя типа символов, который будет использоваться для отображения геометрических примитивов точечного слоя, в середине линии и в центре полигона для слоев линий и полигонов соответственно. Тип значения – строка. Тип символов должен быть объявлен в том же файле описания карты, в котором он используется.

TEMPLATE [filename]	Путь к файлу, содержащему шаблон для вывода атрибутивной информации результатов пространственного поиска по геометрическим примитивам, являющимся членами класса. Тип значения – строка.
NAME [string]	Имя класса геометрических примитивов. Тип значения – строка. Используется в качестве надписей к ключам легенды при создании легенды

LEGEND – этот объект определяет свойства легенды карты. Атрибуты этого объекта позволяют определить размеры ключей легенды, расстояния между ними, цвет фона легенды, прозрачность фона легенды, а также используемый шрифт для отображения надписей. Легенда отображает все классы активных слоев карты. При этом цвета ключей карты соответствуют цветам, указанным в определении соответствующего класса. Этот объект может быть объявлен внутри объекта *MAP*.

Таблица 7 - Атрибуты объекта LEGEND

Атрибут	Описание
IMAGECOLOR [r] [g] [b]	Определяет цвет фона легенды в формате RGB. Тип значения – целое от 0 до 255.
OUTLINECOLOR [r] [g] [b]	Определяет цвет границы ключей легенды в формате RGB. Тип значения – целое от 0 до 255.
KEYSIZE [x] [y]	Определяет размеры ключа легенды в пикселях. Тип значения – целое без знака. По умолчанию устанавливается 25 на 15 пикселей.
KEYSPACING [x][y]	Определяет расстояния между ключами легенды и ключом легенды и надписью. Тип значения – целое без знака. x – определяет расстояние между ключом и надписью для этого ключа. y – расстояние между ключами по вертикали. По умолчанию устанавливается 6 на 6.
TRANSPARENT [on off]	Устанавливает прозрачность фона легенды. Тип значения – строка. По умолчанию прозрачность фона легенды отключена.

MINIMAP – этот объект отвечает за отображение миникарты. Атрибуты этого объекта определяют размеры объемлющего прямоугольника миникарты, цвет линий на ней, прозрачность и размеры миникарты в пикселях, минимальный и максимальный размер рамки, а также символ для отображения текущего центра карты в случае, когда рамка становится слишком маленькой. Этот объект может быть объявлен внутри объекта *MAP*.

Таблица 8 - Атрибуты объекта MINIMAP

Атрибут	Описание
COLOR [r] [g] [b]	Определяет цвет рамки, которая обозначает текущую видимую часть карты. Тип значения – целое от 0 до 255. Если указано значение -1, то рамка не рисуется, содержимое рамки заполняется белым цветом с пятидесятипроцентной прозрачностью.
EXTENT [minx] [miny] [maxx] [maxy]	Размеры объемлющего прямоугольника. Тип значения – double. Используется наряду с параметром <i>EXTENT</i> объекта <i>MAP</i> для

	вычисления положения и размеров рамки. Этот атрибут может быть опущен, в этом случае используется значение атрибута <i>EXTENT</i> объекта <i>MAP</i> .
IMAGE [filename]	Абсолютный к путь к файлу, содержащему изображение миникарты. Тип значения – строка. На это изображение накладывается изображение рамки.
MARKER [string]	Имя типа символов, который будет использован для отображения положения текущего центра карты в том случае, если размер рамки становится слишком маленьким. Тип значения – строка. Минимальный размер рамки определяется атрибутом <i>MINBOXSIZE</i> класса <i>MINIMAP</i> .
MINBOXSIZE [integer]	Определяет минимальный размер рамки в пикселях. Тип значение – целое без знака. В случае, если размер рамки меньше этого значения для отображения текущего положения центра карты используется символ, тип которого определен значением атрибута <i>MARKER</i> объекта <i>MINIMAP</i> . По умолчанию устанавливается 3.
OUTLINECOLOR [r] [g] [b]	Определяет цвет границы миникарты в формате RGB. Тип значения – целое от 0 до 255.
SIZE [x] [y]	Размер базового изображения в пикселях. Тип значения – целое без знака.

SCALEBAR – этот объект отвечает за отображение линейки масштаба. Атрибуты этого объекта позволяют определить цвет фона линейки масштаба, прозрачность фона, размеры линейки, количество интервалов, единицы измерения длин, стиль отображения линейки масштаба, стиль подписей. Этот объект может быть объявлен внутри объекта *MAP*.

Таблица 9 - Атрибуты объекта **SCALEBAR**

Атрибут	Описание
BACKGROUNDCOLOR [r] [g] [b]	Определяет цвет фона элементов линейки масштаба в формате RGB. Тип значения – целое от 0 до 255.
COLOR [r] [g] [b]	Определяет цвет элементов линейки масштаба и надписей в формате RGB. Тип значения – целое от 0 до 255.
IMAGECOLOR [r] [g] [b]	Определяет цвет фона изображения линейки масштаба в формате RGB. Тип значения – целое от 0 до 255.
INTERVALS [integer]	Задаёт количество интервалов на линейке масштаба. Тип значения – целое без знака. Определяет количество интервалов, на которое разбивается линейка масштаба. На границе интервалов отображается надпись, указывающая на расстояние от начала линейки масштаба до границы интервалов в единицах измерения, определенными значением атрибута <i>UNITS</i> объекта <i>SCALEBAR</i> .
SIZE [x] [y]	Размер изображения линейки масштаба в пикселях. Тип значения – целое без знака.
STYLE [integer]	Задаёт стиль для отображения линейки масштаба. Тип значения – 0 или 1. По умолчанию устанавливается 0.

	 <p style="text-align: center;">Рисунок Б.0.1 - Тип линейки масштаба 0</p>  <p style="text-align: center;">Рисунок Б.0.2 - Тип линейки масштаба 1</p>
TRANSPARENT [on off]	Устанавливает прозрачность фона линейки масштаба. Тип значения – строка. По умолчанию прозрачность фона отключена.
UNITS [kilometers meters]	Определяет единицу измерения, используемую для создания надписей, на границе сегментов. Тип значения – строка.

LABEL – Этот объект отвечает за отображение надписей. Он используется при создании миникарты и линейки масштаба. Также он позволяет создавать надписи на самой карте. Атрибуты этого объекта позволяют определить цвет фона надписи, цвет текста надписи, прозрачность фона надписи, размер надписи, положение надписи относительно изображения, на котором она отображается, угол, под которым она отображается, используется ли сглаживание лестничного эффекта. Этот объект может быть объявлен внутри объектов *SCALEBAR*, *MINIMAP*, *MAP*.

Таблица 10 - Атрибуты объекта LABEL

Атрибут	Описание
ANGLE [integer]	Этот атрибут определяет угол поворота надписи в градусах. Тип значения – целое без знака. Точка, определяемая атрибутом <i>OFFSET</i> объекта <i>LABEL</i> , является центром поворота.
BACKGROUNDCOLOR [r] [g] [b]	Определяет цвет фона надписи в формате RGB. Тип значения – целое от 0 до 255.
PADDING [integer]	Размер буферной зоны вокруг надписи в пикселях. Тип значения – целое без знака.
COLOR [r] [g] [b]	Определяет цвет надписи в формате RGB. Тип значения – целое от 0 до 255.
OFFSET [x] [y]	Смещение в пикселях относительно изображения, содержащего надпись. Тип значения – целое без знака. В указанную точку изображения, содержащего надпись, помещается верхний левый угол надписи. Это же точка служит центром поворота на угол, указанный атрибутом <i>ANGLE</i> объекта <i>LABEL</i> .
OUTLINECOLOR [r] [g] [b]	Определяет цвет линии толщиной в один пиксель вокруг каждого символа надписи в формате RGB. Тип значения – целое от 0 до 255.
SIZE [tiny small medium large giant]	Задаёт размер для стандартного системного растрового шрифта, используемого для создания надписи. Тип значения – строка. Любой растровый шрифт в Linux содержит 5 стандартных размеров шрифта, определяемых этим параметром.

WRAP [string]	Определяет, какая последовательность символов используется в качестве символа перевода строки. Тип значения – строка. По умолчанию используется 0x0A.
---------------	---

SYMBOL – этот объект отвечает за отображение символов на карте. В файле описания карты может быть объявлено множество символов, ссылка на каждый конкретный символ производится по его имени. Эти объекты используются для отображения точечных слоев, для установки символов на слоях линий, а также для установки символов на миникарте. Атрибуты этого объекта определяют тип, размеры, цвет, прозрачность символа. В качестве символа может быть использовано графическое изображение в формате JPEG или PNG. Этот объект может быть объявлен внутри объекта *MAP*.

Таблица 11 - Атрибуты объекта **SYMBOL**

Атрибут	Описание
FILLED [true false]	Определяет, заполнен ли символ. Тип значения – строка. Цвет, которым заполняется символ, устанавливается объектом, который использует данный символ.
IMAGE [filename]	Путь к файлу, содержащему графическое изображение в формате JPEG или PNG. Тип значения – строка. Графическое изображение, хранимое в указанном файле, используется для отображения символа.
NAME [string]	Имя типа символов. Тип значения – строка. Используется в качестве уникального идентификатора типа символов, на который ссылаются другие объекты файла описания карты для указания того, какой тип символов необходимо использовать.
TRANSPARENTCOLOR [r] [g] [b]	Указывает на цвет, который надо считать прозрачным для символов, которые для своего отображения используют графическое изображение формата GIF и JPEG, хранимое в файле, определенным атрибутом <i>NAME</i> . Тип значения – целое от 0 до 255.
TYPE [circle triangle square cross]	Определяет тип символа. Тип значения – строка. circle – круг triangle – треугольник square – квадрат cross - крест

METADATA – этот объект служит для хранения произвольных метаданных для объектов *MAP*, *LAYER*, *CLASS*. Атрибутами этого объекта являются значения метаданных. Этот объект может быть объявлен внутри объектов *CLASS*, *LAYER*, *MAP*.

Приложение В. Описание объектно-ориентированного программного интерфейса

В.1 Общее описание

В данный момент создан объектно-ориентированный программный интерфейс, предоставляющий доступ к системе хранения пространственной информации и метаданных и системе визуализации хранимой пространственной информации. Этот интерфейс представляет собой множество классов, определенных и реализованных на языке Perl. Эти классы можно разделить на три группы.

Первая группа – классы для работы с пространственными данными в формате ESRI Shapefile. Методы экземпляров этих классов позволяют осуществлять чтение и запись файлов этого формата, изменять заголовки этих файлов, добавлять и удалять геометрические примитивы, хранимые в этих файлах, получать доступ к отдельным геометрическим примитивам, хранимым в этих файлах. Отдельные геометрические примитивы могут также быть модифицированы. К ним может быть получен доступ на уровне линий, составляющих примитив, и точек, составляющих линии. Для доступа к хранимой атрибутивной информации геометрических примитивов используется

Вторая группа – классы для работы с файлами описания карты. Методы экземпляров этих классов позволяют загружать и сохранять эти файлы, получать доступ к отдельным объектам, хранимым в этом файле, модифицировать значения любых атрибутов объектов файла описания карты. Каждый объект файла описания карты, представлен отдельным классом, экземпляры которого позволяют модифицировать значения атрибутов объекта, и обладают рядом связанными с семантикой объекта методов. Эти методы отвечают за создание, добавление, удаление объектов, получение ссылок на объекты, отрисовку объектов.

Третья группа – ряд классов использующихся во вспомогательных целях. Экземпляры этих классов являются контейнерами данных определенного формата и в основном не имеют методов. Экземпляры этих классов используются в основном для указания параметров вызовов методов.

Экземпляры классов представляют собой ссылки на хеши Perl.

`my $var = $instance->{attribut_name};` - пример чтения значения атрибута экземпляра класса.

`$instance->{attribut_name} = $var;` - пример записи значения атрибута экземпляра класса.

`$instance->metod(LIST)` - пример вызова метода экземпляра класса.

`my $instance = new CLASS(LIST)` - пример вызова конструктора класса.

В.2 Вспомогательные классы

Color – простой класс-контейнер экземпляры которого хранят значение цвета в формате RGB

Таблица 12 - Конструктор класса Color

Конструктор	Описание
new	Создает экземпляр класса. Инициализирует атрибуты компонент цвета нулевыми значениями.

Таблица 13 - Атрибуты экземпляра класса Color

Атрибут	Описание
red	Значение компоненты красного цвета. Тип значения – целое от 0 до 255.
green	Значение компоненты зеленого цвета. Тип значения – целое от 0 до 255.
blue	Значение компоненты синего цвета. Тип значения – целое от 0 до 255.

Таблица 14 – Методы экземпляра класса Color

Метод	Описание
getColor()	Возвращает восстановленный из компонент цвет как десятичное число.
getHTMLColor()	Возвращает восстановленный из компонент цвет в формате принятом для указания произвольных цветов в HTML.

Point – простой класс-контейнер для хранения координат точки на плоскости.

Таблица 15 - Конструктор класса Point

Конструктор	Описание
new	Создает экземпляр класса. Инициализирует атрибуты нулевыми значениями.

Таблица 16 - Атрибуты экземпляра класса Point

Атрибут	Описание
x	Абсцисса точки. Тип значения – double.
y	Ордината точки. Тип значения – double.

Rect – простой класс-контейнер, предназначенный для хранения информации об объемлющих прямоугольниках.

Таблица 17 - Конструктор класса Rect

Конструктор	Описание
new	Создает экземпляр класса. Инициализирует атрибуты нулевыми значениями.

Таблица 18 - Атрибуты экземпляра класса Rect

Атрибут	Описание
leftTop	Условная левая верхняя точка объемлющего прямоугольника. Тип значения – Point.
rightBottom	Условная правая нижняя точка объемлющего прямоугольника. Тип значения – Point.

	– <i>Point</i> .
--	------------------

Image – класс отвечающий за хранение растрового изображения, за его сохранение, доступ к пикселям изображения.

Таблица 19 - Конструктор класса **Image**

Конструктор	Описание
new	Создает экземпляр класса. Инициализирует атрибуты нулевыми значениями.

Таблица 20 - Атрибуты экземпляра класса **Image**

Атрибут	Описание
height	Высота растрового изображения в пикселях. Тип значения – целое без знака.
width	Ширина растрового изображения в пикселях. Тип значения – целое без знака.
type	Графический формат для сохранения изображения в файл. Тип значения – PNG или JPEG.

Таблица 21 – Методы экземпляра класса **Image**

Метод	Описание
save(file => string)	Сохраняет изображение в файл, имя которого определяется параметром file. В случае ошибки возвращает неопределенное значение. Тип графического формата определяется атрибутом <i>type</i> .
setColor(point => <i>Point</i> , color => <i>Color</i>)	Устанавливает цвет пикселя, определенного параметром point в значение, определенное параметром color. В случае ошибки возвращает неопределенное значение.
getColor(point => <i>Point</i>)	Возвращает цвет пикселя, определенного параметром point. Тип возвращаемого значения – <i>Color</i> . В случае ошибки возвращает неопределенное значение.

В.3 Классы для работы с файлами формата ESRI Shapefile

ShapeFile – класс, экземпляры которого предоставляют методы для чтения и записи файлов формата ESRI Shapefile, методы для добавления и удаления геометрических примитивов.

Таблица 22 - Конструктор класса **ShapeFile**

Конструктор	Описание
new	Создает экземпляр класса.

Таблица 23 - Атрибуты экземпляра класса **ShapeFile**

Атрибут	Описание
fileName	Абсолютный путь к файлу формата ESRI Shapefile. Тип значения – с

	трока.
shapesNumber	Возвращает количество геометрических примитивов, содержащихся в файле формата ESRI Shapefile. Тип значения – integer.
type	Возвращает тип полигонов хранимых в файле формат ESRI Shapefile. Тип значения – строка.
bounds	Возвращает объемлющий прямоугольник, содержащий в себе все геометрические примитивы, хранимые в файле формата ESRI Shapefile. Тип значения – <i>Rect</i> .

Таблица 24 – Методы экземпляра класса ShapeFile

Метод	Описание
open(file => string)	Открывает файл формата ESRI Shpaefile. В случае ошибки возвращает неопределенное значение.
save(file => string)	Сохраняет файл формата ESRI Shapefile. В случае ошибки возвращает неопределенное значение.
getShape(index => integer)	Возвращает объект, представляющий геометрический примитив, хранимый в файле формата ESRI Shapefile, под номером, определенным параметром index. Тип возвращаемого значения – <i>Shape</i> . В случае ошибки возвращает неопределенное значение. index – целое число от 0 до <i>shapesNuber</i> .
addShape(shape => <i>Shape</i>)	Добавляет новый геометрический примитив, представленный параметром shape в файл формата ESRI Shapefile. Примитив добавляется после всех существующих примитивов.
deleteShape(index => integer)	Удаляет объект, представляющий геометрический примитив, хранимый в файле формата ESRI Shapefile, под номером, определенным параметром index. Тип возвращаемого значения – <i>Shape</i> . index – целое число от 0 до <i>shapesNuber</i>

Shape – класс представляющий геометрические примитивы, хранимые в файле формата ESRI Shapefile. Каждый геометрический примитив состоит из линий, которые определяются своими узловыми точками. Если геометрический примитив является точкой, то он представляется в виде вырожденной линии, начало и конец которой совпадают. Методы экземпляров этого класса предоставляют средства для доступа к линиям, составляющим геометрический примитив, средства для добавления и удаления линий, средства для определения пересечений с другими геометрическими примитивами.

Таблица 25 - Конструктор класса Shape

Конструктор	Описание
new	Создает экземпляр класса.

Таблица 26 - Атрибуты экземпляра класса Shape

Атрибут	Описание
bounds	Возвращает объемлющий прямоугольник для геометрического примитива.

	Тип значения – <i>Rect</i> .
linesNumber	Возвращает количество линий, из которых состоит геометрический примитив. Тип значения – <i>integer</i> .
type	Возвращает тип геометрического примитива. Тип значения – строка.

Таблица 27 – Методы экземпляра класса *Shape*

Метод	Описание
addLine(line => <i>Line</i>)	Добавляет новую линию к линиям, составляющим геометрический примитив. Линия, которую надо добавить определяется параметром <i>line</i> . Линия добавляется после всех уже существующих линий.
deleteLine(index => <i>integer</i>)	Удаляет линию из множества линий, составляющих геометрический примитив. Параметр <i>index</i> определяет линию, которую необходимо удалить. В случае ошибки возвращает неопределенное значение. <i>index</i> – целое число от 0 до <i>linesNumber</i> .
getLine(index => <i>integer</i>)	Возвращает указанную параметром <i>index</i> линию из множества линий составляющих геометрический примитив. Тип возвращаемого значения – <i>Line</i> . В случае ошибки возвращает неопределенное значение. <i>index</i> – целое число от 0 до <i>linesNumber</i> .
copy(shape => <i>Shape</i>)	Копирует параметры указанного параметром <i>shape</i> геометрического примитива, другими словами создает копию указанного геометрического примитива из текущего примитива. Параметры текущего геометрического примитива не сохраняются. В случае ошибки возвращает неопределенное значение.
intersects(shape => <i>Shape</i>)	Определяет наличие пересечений между двумя геометрическими примитивами. В качестве критерия наличия пересечения используется наличие хотя бы одной вершины одного геометрического на границе или внутри другого. Такой критерий работает и для точечных примитивов, которые представлены в виде вырожденной линии с совпадающими началом и концом. В случае отсутствия пересечения возвращается неопределенное значение.

Line – класс представляющий линии, из которых состоит геометрический примитив. Линии состоят из двух или более точек, представленных вышеописанным классом *Point*. Методы экземпляра этого класса предоставляют средства для добавления и удаления точек, составляющих линию, средства для доступа к точкам, составляющим линию.

Таблица 28 - Конструктор класса *Line*

Конструктор	Описание
new	Создает экземпляр класса.

Таблица 29 - Атрибуты экземпляра класса *Line*

Атрибут	Описание
pointsNumber	Возвращает количество точек, составляющих линию. Тип значения –

	integer.
--	----------

Таблица 30 – Методы экземпляра класса Line

Метод	Описание
addPoint(point => Point)	Добавляет точку к множеству точек, составляющих линию. Точка для добавления определяется параметром point. Точка добавляется после всех существующих точек.
getPoint(index => integer)	Возвращает указанную параметром index точку. Тип возвращаемого значения – Point. В случае ошибки возвращается неопределенное значение. index – целое от 0 до pointsNumber.
deletePoint(index => integer)	Удаляет точку из множества точек, составляющих линию. Параметр index определяет точку, которую необходимо удалить. index – целое от 0 до pointsNumber.

В.4 Классы для работы с объектами файла описания карты

Map – класс представляющий собой объект MAP файла описания карты.

Таблица 31 - Конструктор класса Map

Конструктор	Описание
new	Создает экземпляр класса.

Таблица 32 – Атрибуты экземпляра класса Map

Атрибут	Описание
bounds	Возвращает объемлющий прямоугольник карты. Тип значения – Rect.
height	Размер изображения карты по вертикали в пикселях. Тип значения – integer.
width	Размер изображения карты по горизонтали в пикселях. Тип значения – integer.
imagecolor	Цвет фона карты. Тип возвращаемого значения – Color.
imagetype	Тип графического формата, используемого для сохранения изображения карты, легенды, миникарты, линейки масштаба. Тип значения – строка. Доступные значения: PNG, JPEG.
legend	Возвращает экземпляр класса Legend, представляющего объект файла описания карты LEGEND. Тип возвращаемого значения – Legend.
name	Возвращает значение атрибута NAME объекта файла описания карты MAP. Тип значения – строка.
layersNumber	Возвращает количество объявленных в объекте MAP файла описания карты объектов LAYER. Тип значения – integer.
minimap	Возвращает экземпляр класса Minimap, представляющего объект файла описания карты MINIMAP. Тип значения – Minimap.
scalebar	Возвращает экземпляр класса Scalebar, представляющего объект файла описания карты Scalebar. Тип значения – Scalebar.

datapath	Возвращает абсолютный путь к директории, содержащей файлы формата ESRI Sapefile, использующиеся для создания слоев карты. Тип значения – строка.
transparent	Показывает, установлена ли прозрачность фона изображения карты. Тип значения – integer. Возвращает 0, если прозрачность не установлена.
metadata	Возвращает экземпляр класса <i>Metadata</i> , представляющего объект файла описания карты <i>METADATA</i> . Тип значения – <i>Metadata</i> .
units	Возвращает тип единицы измерения, в которой заданы координаты точек, составляющих геометрические примитивы слоев, составляющих карту. Тип значения – строка. Доступные значения – meters, kilometers, dd (digital degrees).

Таблица 33 – Методы экземпляра класса Map

Метод	Описание
draw()	Рисует активные слои карты в соответствии с информацией, содержащейся в файле описания карты. Тип возвращаемого значения – <i>Image</i> . В случае ошибки возвращает неопределенное значение. Для создания изображения карты используется информация о слоях карты, о классах каждого слоя и текущий масштаб карты. На основании определения классов из множества геометрических примитивов, составляющих слой, выделяются классы, и рисуются, если текущее значение масштаба позволяет это сделать. Класс геометрических примитивов рисуется в соответствии со своим описанием. Для рисования используется информация о цвете фона и границ геометрических примитивов, а также, возможно, информация о символах, которые могут быть дополнительно использованы для отображения геометрических примитивов, принадлежащих определенному классу.
getLayer(index => integer)	Возвращает экземпляр класса <i>Layer</i> , определенный параметром <i>index</i> , представляющий объект файла описания карты <i>LAYER</i> . Тип возвращаемого значения – <i>Layer</i> . В случае ошибки возвращается неопределенное значение. <i>index</i> – целое от 0 до <i>layersNumber</i> .
layerUp(index => integer)	Изменяет порядок отображения слоев. Передвигает слой, указанный параметром <i>index</i> вверх на одну позицию. Чем слой выше, тем позже рисуются геометрические примитивы, его составляющие. В случае ошибки возвращает неопределенное значение
layerDown(index => integer)	Изменяет порядок отображения слоев. Передвигает слой, указанный параметром <i>index</i> вниз на одну позицию. Чем слой ниже, тем позже рисуются геометрические примитивы, его составляющие. В случае ошибки возвращает неопределенное значение
query(shape => <i>Shape</i>)	Осуществляет пространственный поиск. Тип возвращаемого значения – <i>Map</i> . В случае неудачного поиска возвращает неопределенное значение. В состав объекта <i>Map</i> будут входить только те слои, классы и, геометрические примитивы которых имеют пересечение с определенным параметром <i>shape</i> геометрическим примитивом. В классы будут входить только те примитивы, которые имеют

	пересечение с указанным примитивом.
save(file => string)	Сохраняет экземпляр класса Map в файл формата описания карты. В случае ошибки возвращает неопределенное значение. Файл для сохранения определяется параметром file. Если файл уже существует – то он перезаписывается.
open(file => string)	Открывает файл формата описания карты, производит его разбор, инициализирует значениями из файла атрибуты экземпляры класса Map в случае, если файл описания карты имеет правильный синтаксис. В случае ошибки возвращает неопределенное значение.

Layer – класс представляющий объект *LAYER* файла описания карты.

Таблица 34 - Конструктор класса Layer

Конструктор	Описание
new	Создает экземпляр класса.

Таблица 35 - Атрибуты экземпляра класса Layer

Атрибут	Описание
classfield	Представляет собой значение атрибута <i>CLASSFIELD</i> объекта <i>LAYER</i> файла описания карты. Тип значения – строка. Используется для создания классов геометрических примитивов среди геометрических примитивов слоя.
data	Представляет собой значение атрибута <i>DATA</i> объекта <i>LAYER</i> файла описания карты. Тип значения – строка. Значение этого атрибута определяет абсолютный путь к файлу формата ESRI Shapefile, содержащему геометрические примитивы этого слоя.
header	Представляет собой значение атрибута <i>HEADER</i> объекта <i>LAYER</i> файла описания карты. Тип значения – строка. Значение этого атрибута определяет абсолютный путь к шаблону, который будет использоваться в верхней части страницы для представления результатов поиска по геометрическим примитивам, составляющим слой карты.
footer	Представляет собой значение атрибута <i>FOOTER</i> объекта <i>LAYER</i> файла описания карты. Тип значения – строка. Значение этого атрибута определяет абсолютный путь к шаблону, который будет использоваться в нижней части страницы для представления результатов поиска по геометрическим примитивам, составляющим слой карты.
template	Представляет собой значение атрибута <i>TEMPLATE</i> объекта <i>LAYER</i> файла описания карты. Тип значения – строка. Значение этого атрибута определяет абсолютный путь к шаблону, который будет использоваться в основной части страницы для представления результатов поиска по геометрическим примитивам, составляющим слой карты. Значение этого атрибута используется в том случае, если это значение не установлено для классов геометрических примитивов, составляющих этот слой.
maxscale	Представляет собой значение атрибута <i>MAXSCALE</i> объекта <i>LAYER</i> файла описания карты. Тип значения – double. Значение этого атрибута

	используется для определения максимального масштаба, при котором отображаются геометрические примитивы, составляющие данный слой.
minscale	Представляет собой значение атрибута <i>MINSCALE</i> объекта <i>LAYER</i> файла описания карты. Тип значения – double. Значение этого атрибута используется для определения минимального масштаба, при котором отображаются геометрические примитивы, составляющие данный слой.
name	Представляет собой значение атрибута <i>MAXSCALE</i> объекта <i>LAYER</i> файла описания карты. Тип значения – double. Значение этого атрибута используется для определения максимального масштаба, при котором отображаются геометрические примитивы, составляющие данный слой.
status	Представляет собой значение атрибута <i>STATUS</i> объекта <i>LAYER</i> файла описания карты. Тип значения – integer. Возвращает ноль, если слой карты невиден. Единицу, если слой виден. Двойку если слой виден по умолчанию.
type	Представляет собой значение атрибута <i>TYPE</i> объекта <i>LAYER</i> файла описания карты. Тип значения – строка. Значение этого атрибута содержит тип геометрических примитивов, составляющих слой карты. Возможные значения: point, line, polygon.
classesNumber	Возвращает количество классов для этого слоя, объявленных в файле описания карты.
metadata	Возвращает экземпляр класса <i>Metadata</i> , представляющего объект файла описания карты <i>METADATA</i> . Тип значения – <i>Metadata</i> .

Таблица 36 – Методы экземпляра класса Layer

Метод	Описание
draw(map => Map)	Рисует геометрические примитивы, составляющие слой карты. Карта задается параметром map. Для рисования используется информация об объемлющем прямоугольнике карты, размерах изображения карты, масштабе. Тип возвращаемого значения – Image. В случае ошибки возвращает неопределенное значение.
getClass(index => integer)	Возвращает экземпляр класса Class, определенного параметром index, представляющего объект файла описания карты CLASS. Тип возвращаемого значения – Class. В случае ошибки возвращается неопределенное значение.
query(shape => Shape)	Осуществляет пространственный поиск. Тип возвращаемого значения – Layer. В объект Layer будут входить только те классы и геометрические примитивы, которые имеют пересечения с геометрическим примитивом, определенным параметром shape. В случае неудачного поиска возвращается неопределенное значение.

Class – класс представляющий объект CLASS файла описания карты.

Таблица 37 - Конструктор класса Class

Конструктор	Описание
new	Создает экземпляр класса.

Таблица 38 - Атрибуты экземпляра класса Class

Атрибут	Описание
backgroundcolor	Представляет собой значение атрибута <i>BACKGROUND</i> объекта <i>CLASS</i> файла описания карты. Тип значения – <i>Color</i> . Значение этого атрибута используется для задания фона геометрических примитивов, составляющих класс.
color	Представляет собой значение атрибута <i>COLOR</i> объекта <i>CLASS</i> файла описания карты. Тип значения – <i>Color</i> . Значение используется для задания границы геометрических примитивов, составляющих класс.
filter	Представляет собой значение атрибута <i>FILTER</i> объекта <i>CLASS</i> файла описания карты. Тип значения – строка. Значение этого атрибута используется для определения класса примитивов на основе их атрибутивной информации.
maxscale	Представляет собой значение атрибута <i>MAXSCALE</i> объекта <i>CLASS</i> файла описания карты. Тип значения – <i>double</i> . Значение этого атрибута используется для определения максимального масштаба, при котором отображаются геометрические примитивы, составляющие класс.
minscale	Представляет собой значение атрибута <i>MINSCALE</i> объекта <i>CLASS</i> файла описания карты. Тип значения – <i>double</i> . Значение этого атрибута используется для определения минимального масштаба, при котором отображаются геометрические примитивы, составляющие класс.
symbol	Представляет собой значение атрибута <i>SYMBOL</i> объекта <i>CLASS</i> файла описания карты. Тип значения – строка. Значение этого атрибута используется для указания определенного в файле описания карты типа символов, используемого для отображения геометрических примитивов, составляющих класс.
template	Представляет собой значение атрибута <i>TEMPLATE</i> объекта <i>CLASS</i> файла описания карты. Тип значения – строка. Значение этого атрибута используется для указания имени файла-шаблона, который будет использоваться для представления результатов пространственного поиска по геометрическим примитивам, составляющим класс.
metadata	Возвращает экземпляр класса <i>Metadata</i> , представляющего объект файла описания карты <i>METADATA</i> . Тип значения – <i>Metadata</i> .
name	Представляет собой значение атрибута <i>NAME</i> объекта <i>CLASS</i> файла описания карты. Тип значения – строка. Значение этого атрибута используется для указания названия класса. Это значение используется в качестве надписи ключа легенды при ее создании.

Таблица 39 – Методы экземпляра класса Class

Метод	Описание
draw(map => Map, layer => Layer)	Рисует геометрические примитивы, составляющие класс. Тип возвращаемого значения – <i>Image</i> . В случае ошибки возвращает неопределенное значение. Для рисования используется значения атрибутов экземпляров классов Map и Layer, определенные параметрами

	map и layer соответственно.
--	-----------------------------

Legend – класс представляющий объект *LEGEND* файла описания карты.

Таблица 40 - Конструктор класса Legend

Конструктор	Описание
new	Создает экземпляр класса.

Таблица 41 - Атрибуты экземпляра класса Legend

Атрибут	Описание
imagecolor	Представляет собой значение атрибута <i>IMAGECOLOR</i> объекта <i>LEGEND</i> файла описания карты. Тип значения – <i>Color</i> . Значение этого атрибута используется для задания цвета фона изображения легенды.
outlinecolor	Представляет собой значение атрибута <i>OUTLINECOLOR</i> объекта <i>LEGEND</i> файла описания карты. Тип значения – <i>Color</i> . Значение этого атрибута используется для задания цвета границы изображения легенды.
width	Определяет ширину ключа на легенде в пикселях. Тип значения – <i>integer</i> .
height	Определяет высоту ключа на легенде в пикселях. Тип значения – <i>integer</i> .
label	Возвращает экземпляр класса <i>Label</i> , представляющего объект файла описания карты <i>LABEL</i> . Тип значения – <i>Label</i> .
transparent	Представляет собой значение атрибута <i>TRANSPARENT</i> объекта <i>LEGEND</i> файла описания карты. Тип значения – <i>integer</i> . Значение этого атрибута используется для включения прозрачности фона изображения легенды. Возвращает истинное значение, если прозрачность включена.
verticalSpacing	Определяет расстояние между ключами карты по вертикали в пикселях. Тип значения – <i>integer</i> .
horizontalSpacing	Определяет расстояние между ключами карты и надписями, а также расстояние между левым краем изображения легенды и ключом легенды в пикселях. Тип значения – <i>integer</i> .

Таблица 42 – Методы экземпляра класса Legend

Метод	Описание
draw(map => Map)	Рисует легенду карты. Тип возвращаемого значения – <i>Image</i> . В случае ошибки возвращает неопределенное значение. Для рисования необходимы значения атрибутов слоев и классов, составляющих карту. Для этого передается экземпляр класса <i>Map</i> , который определен параметром map.

Minimap – класс представляющий объект *MINIMAP* файла описания карты.

Таблица 43 - Конструктор класса *Minimap*

Конструктор	Описание
new	Создает экземпляр класса.

Таблица 44 - Атрибуты экземпляра класса *Minimap*

Атрибут	Описание
color	Представляет собой значение атрибута <i>COLOR</i> объекта <i>MINIMAP</i> файла описания карты. Тип значения – <i>Color</i> . Значение этого атрибута используется для определения цвета рамки, отображающей видимую часть карты.
extent	Представляет собой значение атрибута <i>EXTENT</i> объекта <i>MINIMAP</i> файла описания карты. Тип значения – <i>Rect</i> . Значение этого атрибута используется для задания объемлющего прямоугольника миникарты.
image	Представляет собой значение атрибута <i>IMAGE</i> объекта <i>MINIMAP</i> файла описания карты. Тип значения – строка. Значение этого атрибута используется для указания абсолютного пути к файлу, содержащему изображение, которое используется в качестве фона миникарты.
marker	Представляет собой значение атрибута <i>MARKER</i> объекта <i>MINIMAP</i> файла описания карты. Тип значения – строка. Значение этого атрибута используется для указания имени типа символов, объявленного в файле описания карты, который используется для отображения текущего центра карты в случае, если размер рамки меньше значения атрибута <i>minboxsize</i> .
minboxsize	Представляет собой значение атрибута <i>MINBOXSIZE</i> объекта <i>MINIMAP</i> файла описания карты. Тип значения – <i>integer</i> . Значение этого атрибута используется для указания минимального размера рамки, отображающей текущую видимую часть карты, в пикселях. В случае, если размер рамки меньше этого значения, для отображения текущего центра карты используется символ, тип которого определен значением атрибута <i>marker</i> .
outlinecolor	Представляет собой значение атрибута <i>OUTLINECOLOR</i> объекта <i>MINIMAP</i> файла описания карты. Тип значения – <i>Color</i> . Значение этого атрибута используется для задания цвета границы изображения миникарты.
width	Определяет ширину изображения миникарты в пикселях. Тип значения – <i>integer</i> .
height	Определяет высоту изображения миникарты в пикселях. Тип значения – <i>integer</i> .

Таблица 45 – Методы экземпляра класса *Minimap*

Метод	Описание
draw(map => <i>Map</i>)	Рисует миникарту. Тип возвращаемого значения – <i>Image</i> . В случае ошибки возвращает неопределенное значение. Для рисования необходимы значения масштаба и объемлющего прямоугольника карты. Для этого передается экземпляр класса <i>Map</i> , который определен параметром <i>map</i> .

Scalebar – класс представляющий объект *SCALEBAR* файла описания карты.

Таблица 46 - Конструктор класса **Scalebar**

Конструктор	Описание
new	Создает экземпляр класса.

Таблица 47 - Атрибуты экземпляра класса **Scalebar**

Атрибут	Описание
backgroundcolor	Представляет собой значение атрибута <i>BACKGROUNDCOLOR</i> объекта <i>SCALEBAR</i> файла описания карты. Тип значения – <i>Color</i> . Значение этого атрибута используется для определения цвета фона изображения линейки масштаба.
color	Представляет собой значение атрибута <i>COLOR</i> объекта <i>SCALEBAR</i> файла описания карты. Тип значения – <i>Color</i> . Значение этого атрибута используется для определения цвета, которым отображаются основные элементы линейки масштаба, а также надписи на границе интервалов линейки масштаба.
imagecolor	Представляет собой значение атрибута <i>IMAGECOLOR</i> объекта <i>SCALEBAR</i> файла описания карты. Тип значения – <i>Color</i> . Значение этого атрибута используется для определения цвета, которым отображаются дополнительные элементы линейки масштаба.
intervals	Представляет собой значение атрибута <i>INTERVALS</i> объекта <i>SCALEBAR</i> файла описания карты. Тип значения – <i>integer</i> . Значение этого атрибута определяет количество интервалов на линейке масштаба.
width	Определяет ширину изображения линейки масштаба в пикселях. Тип значения – <i>integer</i> .
height	Определяет высоту изображения линейки масштаба в пикселях. Тип значения – <i>integer</i> .
transparent	Представляет собой значение атрибута <i>TRANSPARENT</i> объекта <i>SCALEBAR</i> файла описания карты. Тип значения – <i>integer</i> . Значение этого атрибута используется для включения прозрачности фона изображения легенды. Возвращает истинное значение, если прозрачность включена.
label	Возвращает экземпляр класса <i>Label</i> , представляющего объект файла описания карты <i>LABEL</i> . Тип значения – <i>Label</i> .
units	Представляет собой значение атрибута <i>UNITS</i> объекта <i>SCALEBAR</i> файла описания карты. Тип значения – <i>integer</i> . Значение этого атрибута определяет единицу измерения, в которой указывается значение масштаба в надписях на границе интервалов линейки масштаба.

Таблица 48 - – Методы экземпляра класса **Scalebar**

Метод	Описание
draw(map => Map)	Рисует линейку масштаба. Тип возвращаемого значения – <i>Image</i> . В случае ошибки возвращает неопределенное значение. Для рисования необходимы

	значения масштаба и объемлющего прямоугольника карты. Для этого передается экземпляр класса <i>Map</i> , который определен параметром <i>map</i> .
--	--

Label – класс представляющий объект *LABEL* файла описания карты.

Таблица 49 - Конструктор класса Label

Конструктор	Описание
new	Создает экземпляр класса.

Таблица 50 - Атрибуты экземпляра класса Label

Атрибут	Описание
angle	Представляет собой значение атрибута <i>ANGLE</i> объекта <i>LABEL</i> файла описания карты. Тип значения – <i>double</i> . Значение этого атрибута определяет угол поворота надписи.
backgroundcolor	Представляет собой значение атрибута <i>BACKGROUNDCOLOR</i> объекта <i>LABEL</i> файла описания карты. Тип значения – <i>Color</i> . Значение этого атрибута определяет цвет фона надписи.
padding	Представляет собой значение атрибута <i>PADDING</i> объекта <i>LABEL</i> файла описания карты. Тип значения – <i>integer</i> . Значение этого атрибута определяет размер буферной зоны вокруг надписи в пикселях. Буферная зона окрашивается в цвет фона надписи.
color	Представляет собой значение атрибута <i>BACKGROUNDCOLOR</i> объекта <i>LABEL</i> файла описания карты. Тип значения – <i>Color</i> . Значение этого атрибута определяет цвет надписи.
offsetX	Определяет смещение надписи по горизонтали в пикселях относительно изображения, на которое она накладывается. Тип значения – <i>integer</i> .
offsetY	Определяет смещение надписи по вертикали в пикселях относительно изображения, на которое она накладывается. Тип значения – <i>integer</i> .
outlinecolor	Представляет собой значение атрибута <i>OUTLINECOLOR</i> объекта <i>LABEL</i> файла описания карты. Тип значения – <i>Color</i> . Значение этого атрибута определяет цвет границы надписи.
size	Представляет собой значение атрибута <i>SIZE</i> объекта <i>LABEL</i> файла описания карты. Тип значения – строка. Значение этого атрибута используется для указания размера растрового шрифта, используемого для рисования надписи.
wrap	Представляет собой значение атрибута <i>WRAP</i> объекта <i>LABEL</i> файла описания карты. Тип значения – <i>Color</i> . Значение этого атрибута определяет последовательность символов используемую для определения начала новых строк в тексте надписи.

Таблица 51 - – Методы экземпляра класса Label

Метод	Описание
draw()	Рисует надпись. Тип возвращаемого значения – <i>Image</i> . В случае ошибки

	возвращает неопределенное значение.
--	-------------------------------------

Symbol – класс представляющий объект *SYMBOL* файла описания карты.

Таблица 52 - Конструктор класса **Symbol**

Конструктор	Описание
new	Создает экземпляр класса.

Таблица 53 - Атрибуты экземпляра класса **Symbol**

Атрибут	Описание
filled	Представляет собой значение атрибута <i>FILLED</i> объекта <i>SYMBOL</i> файла описания карты. Тип значения – integer. Значение этого атрибута показывает надо ли заполнять фон символа. Возвращает истинное значение, если надо.
image	Представляет собой значение атрибута <i>IMAGE</i> объекта <i>SYMBOL</i> файла описания карты. Тип значения – строка. Значение этого атрибута указывает абсолютный путь к файлу, содержащему изображение, используемое для отображения символа.
name	Представляет собой значение атрибута <i>NAME</i> объекта <i>SYMBOL</i> файла описания карты. Тип значения – строка. Значение этого атрибута используется в качестве уникального имени типа символов.
transparentcolor	Представляет собой значение атрибута <i>FILLED</i> объекта <i>SYMBOL</i> файла описания карты. Тип значения – <i>Color</i> . Значение этого атрибута определяет прозрачный цвет в изображении, хранящемся в файле, указанном атрибутом <i>image</i> .
type	Представляет собой значение атрибута <i>TYPE</i> объекта <i>SYMBOL</i> файла описания карты. Тип значения – строка. Значение этого атрибута определяет тип символа.

Приложение Г. Иллюстрации

Г.1 Пользовательская часть

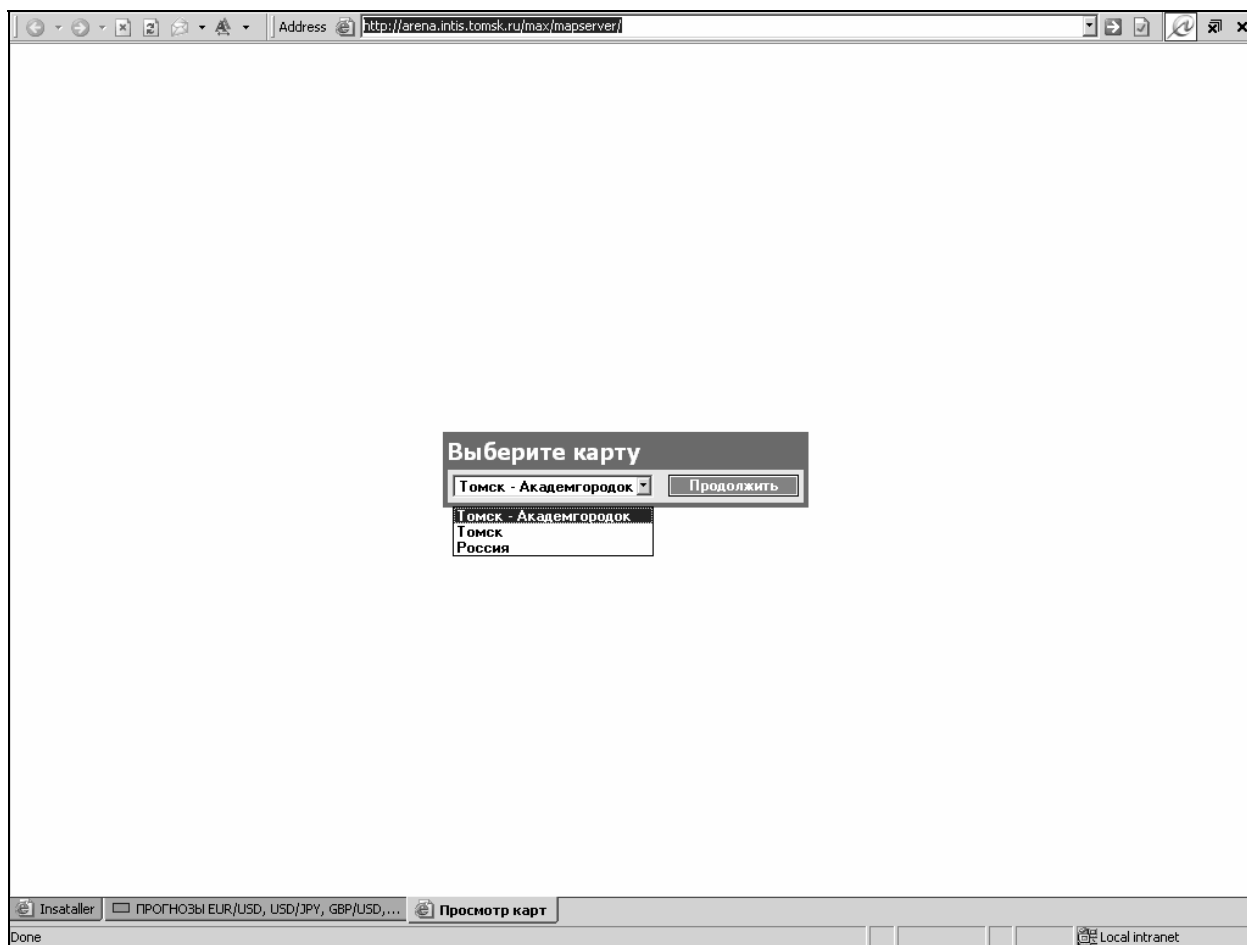


Рисунок Г.1 – Окно выбора карты

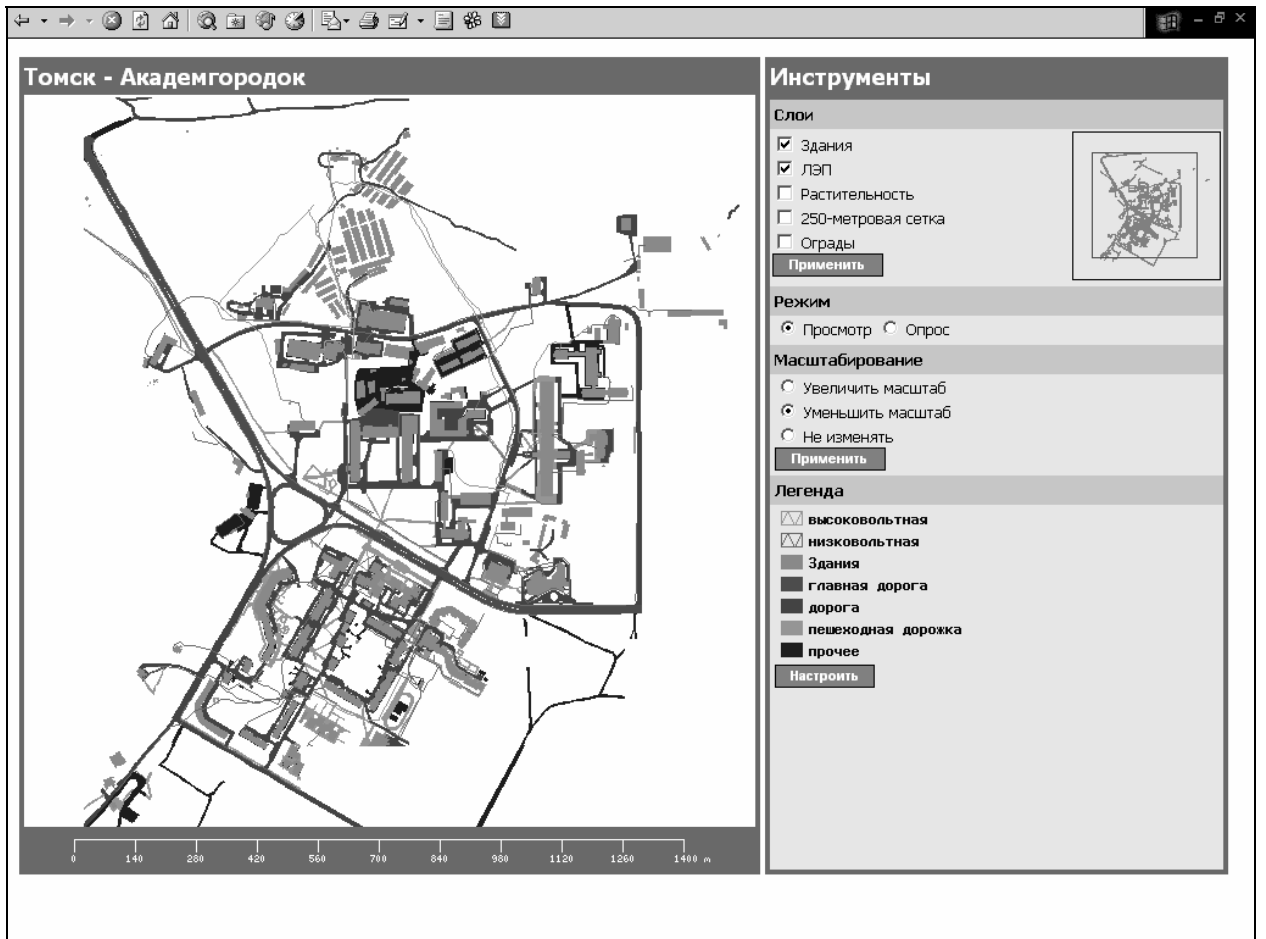


Рисунок Г.2 - Окно работы с картой

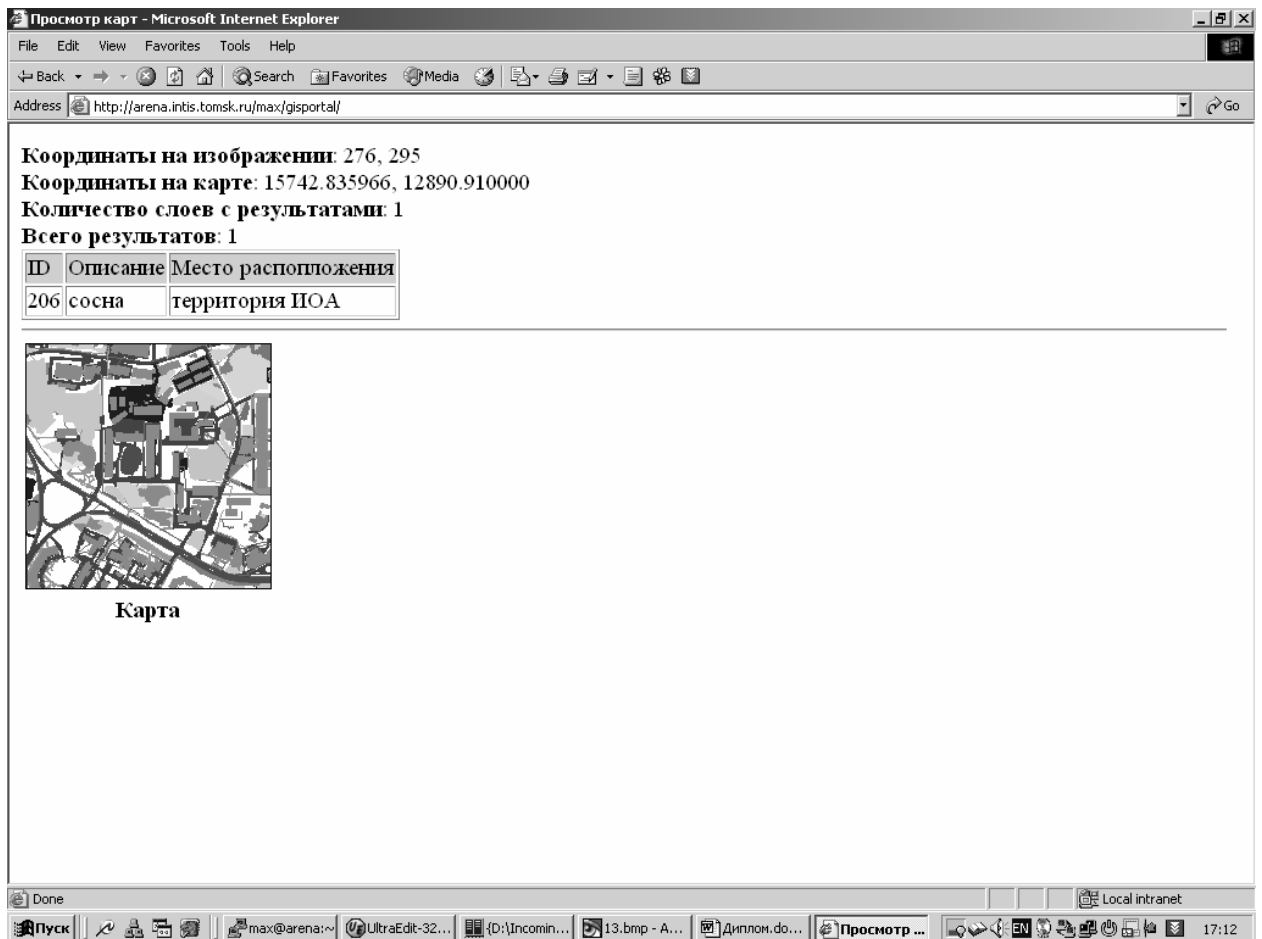


Рисунок Г.3 - Окно результатов пространственного поиска

Г.2 Административная часть

Список карт - Microsoft Internet Explorer

Address http://arena.intis.tomsk.ru/cgi-bin/max/gisportal/admin_viewer.pl

Список карт

Имя файла	Название	Размеры	Единица измерения	Слой
russia.map	Россия <input type="text"/> <input type="button" value="Сохранить"/>	81.86	градус	СЛОИ
		-180.00 180.00		
		42.29		
tomsk.map	Томск <input type="text"/> <input type="button" value="Сохранить"/>	56.54	градус	СЛОИ
		84.90 85.15		
		56.45		
academ.map	Томск - Академгородок <input type="text"/> <input type="button" value="Сохранить"/>	13671.30	метр	СЛОИ
		15003.00 16487.90		
		12105.30		

Local intranet 16:56

Рисунок Г.4 - Окно списка карт

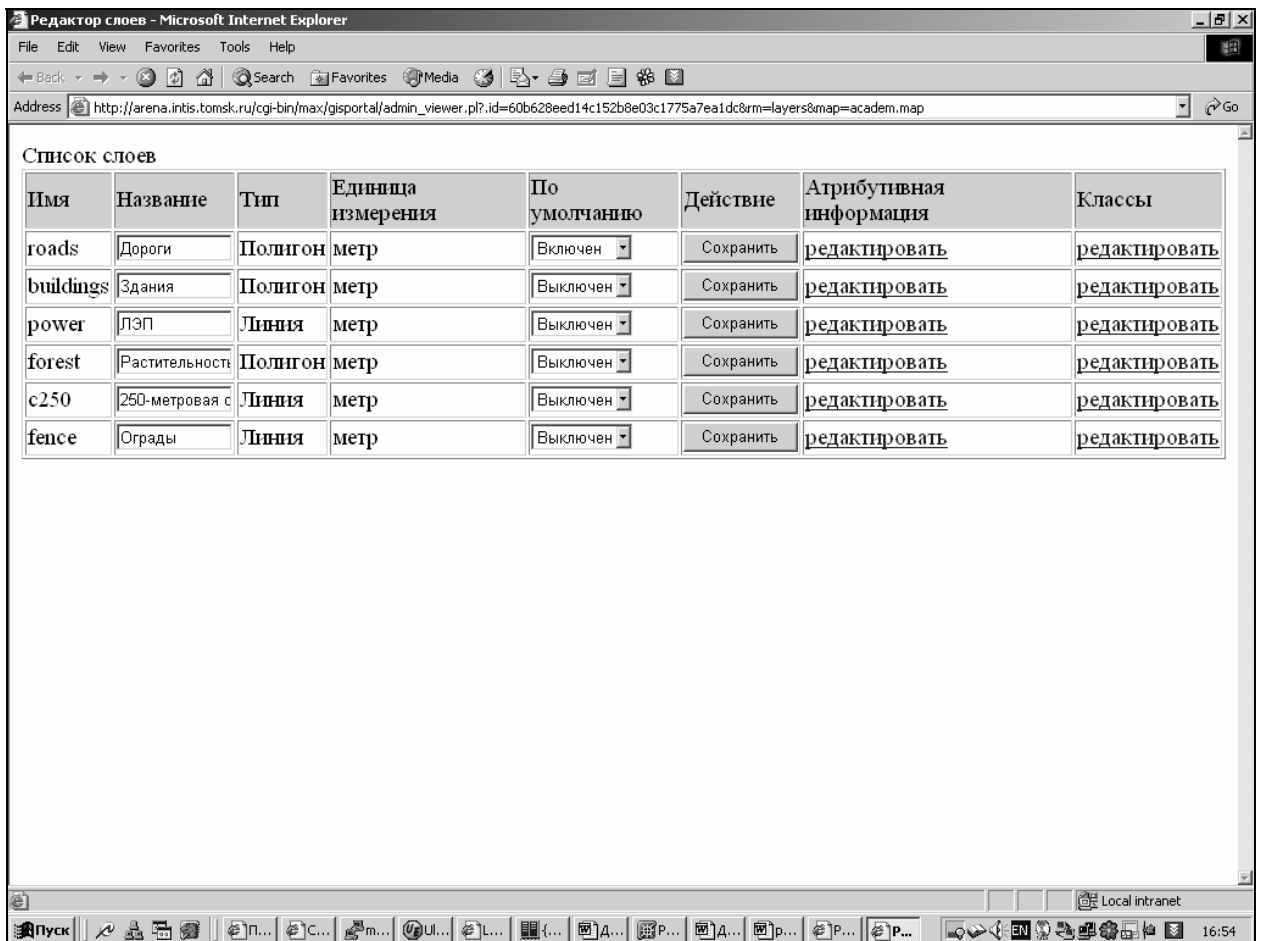


Рисунок Г.5 - Окно редактора слоев

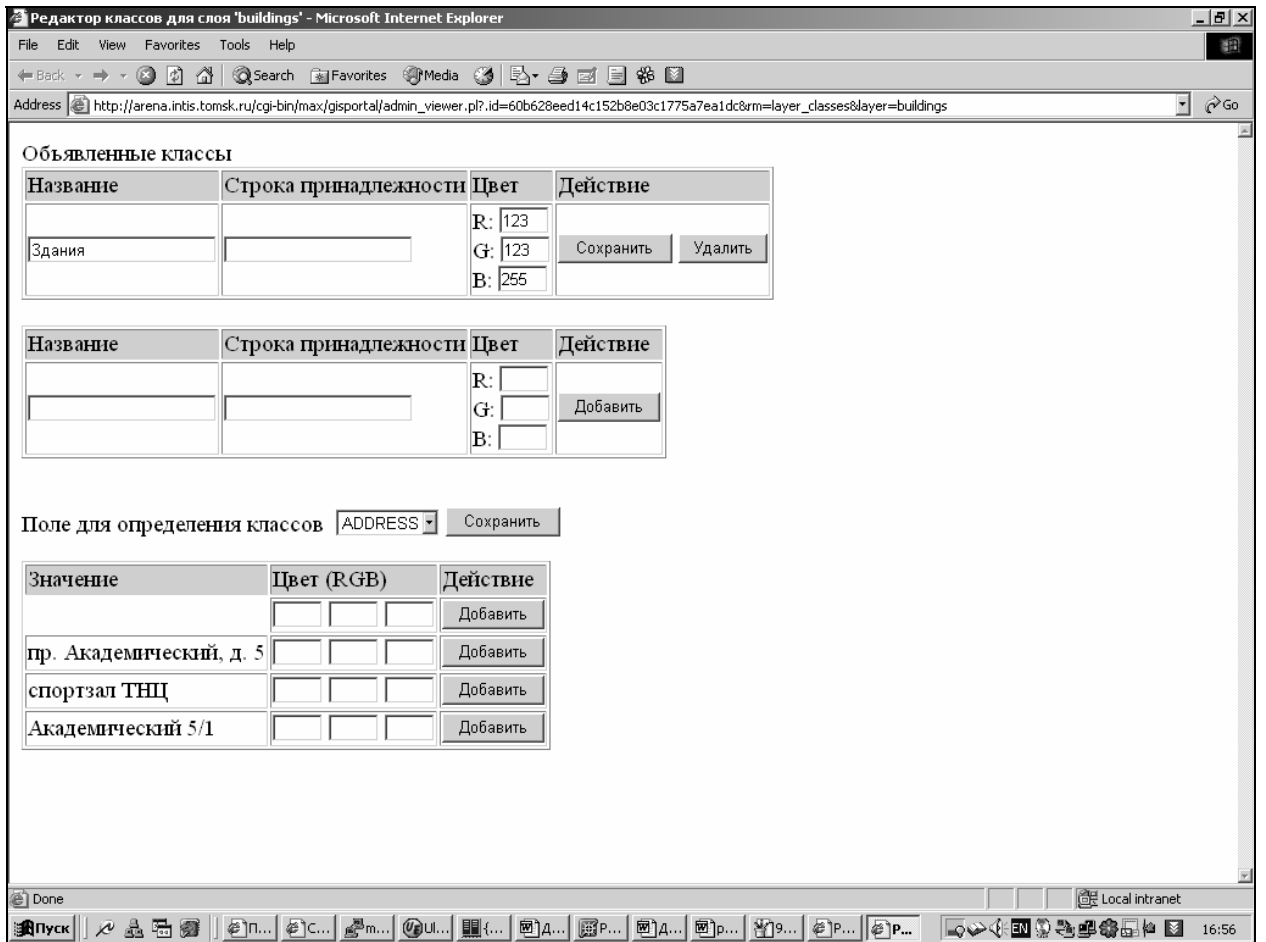


Рисунок Г.6 - Окно редактора классов

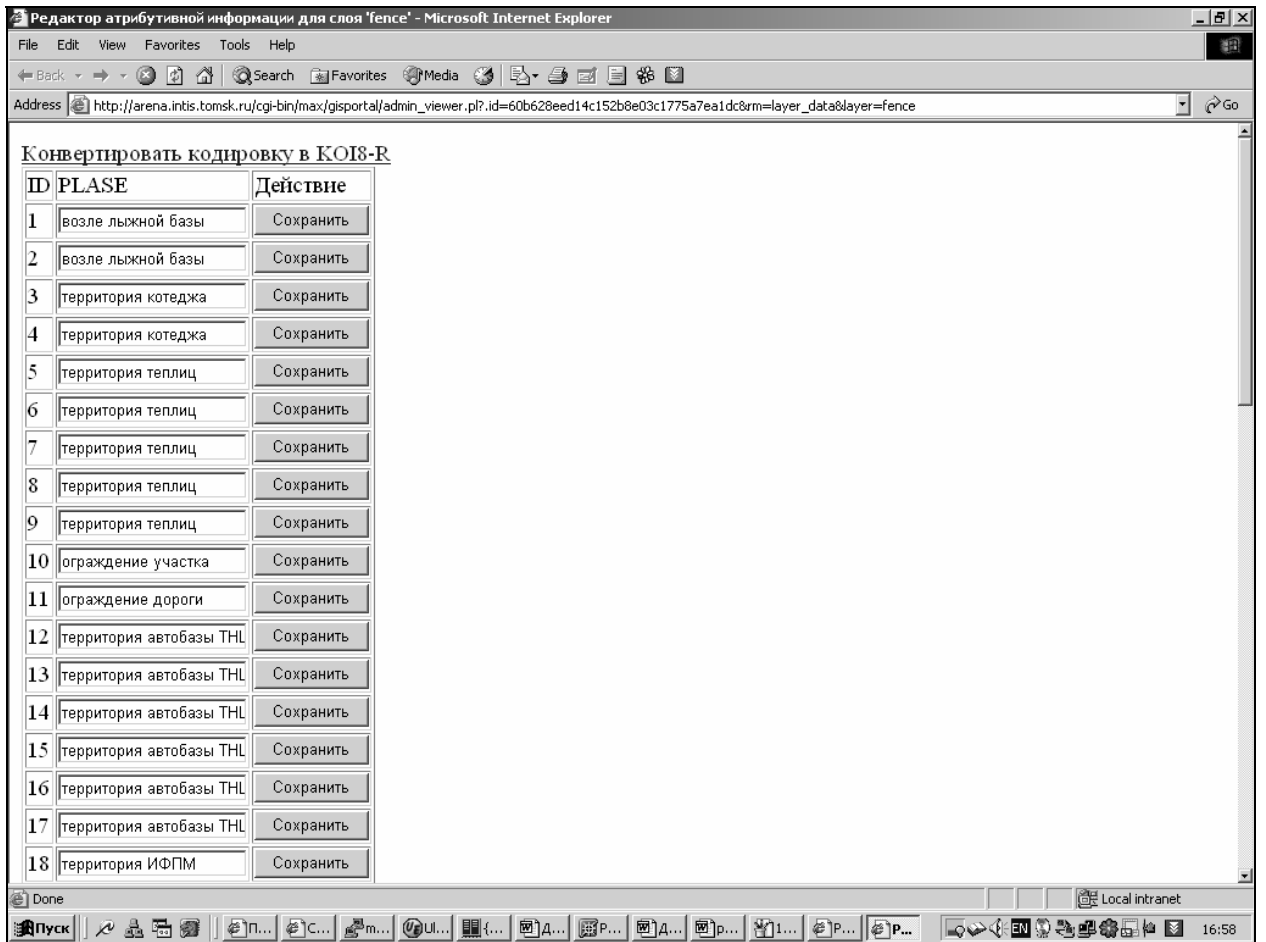


Рисунок Г.7 - Окно редактора атрибутивной информации