

МИНИСТЕРСТВО ОБРАЗОВАНИЯ РЕСПУБЛИКИ БЕЛАРУСЬ
БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
МЕЖДУНАРОДНЫЙ ГОСУДАРСТВЕННЫЙ ЭКОЛОГИЧЕСКИЙ
ИНСТИТУТ им. А.Д. САХАРОВА

Кафедра информационных технологий в экологии и медицине

МОДУЛЬ ГЕОПРОСТРАНСТВЕННОГО АНАЛИЗА И ОРГАНИЗАЦИИ
ВЗАИМОСВЯЗЕЙ ЭЛЕМЕНТАРНЫХ УЧАСТКОВ В СОСТАВЕ
ПРОГРАММНОГО КОМПЛЕКСА ФОРМИРОВАНИЯ И ВЕДЕНИЯ
БАНКА ДАННЫХ АГРОХИМИЧЕСКИХ СВОЙСТВ ПОЧВ

Дипломная работа
Направление специальности
1-40 05 01-07 Информационные системы и технологии (в здравоохранении)

Ероминок Евгений Сергеевич

Научный руководитель:
кандидат физико-математических
наук, доцент П.К. Шалькевич

Допустить к защите
Заведующий кафедрой
_____ И. А. Тавгень
«___» «_____» 2025 г.

Минск, 2025

РЕФЕРАТ

Дипломная работа – 63 страницы, 12 рисунков, 1 таблица, 15 источника.

ГЕОИНФОРМАЦИОННЫЕ СИСТЕМЫ, АГРОХИМИЯ, DRUPAL 10, PHP 8.4, MYSQL, BOOST.GEOMETRY, LEAFLETJS, OPENSTREETMAP, DOCKER, GIT, WATCH API.

Практическая значимость работы заключается в разработке и внедрении модуля геопространственного анализа в рамках банка данных для станций химизации. Решение направлено на автоматизацию хранения, отображения и обработки пространственно-временной информации об элементарных участках сельскохозяйственных земель.

Целью работы является разработка модуля геопространственного анализа, обеспечивающего фиксацию, визуализацию и сравнение геометрических данных участков в разные туры обследования, а также поддержку связей между участками и их предшественниками.

Задачи:

1. Проанализировать деятельность станций химизации и определить ключевые требования к модулю.
2. Спроектировать архитектуру программного решения с учётом работы с геоданными.
3. Реализовать модуль на платформе Drupal 10 с использованием PHP и MySQL.
4. Обеспечить визуализацию участков на карте с применением LeafletJS и OpenStreetMap.
5. Реализовать алгоритм определения пересечений участков между турами на основе Boost.Geometry и метода Вейлера-Азертонна.
6. Обеспечить возможность хранения информации о наследственных связях участков.
7. Провести тестирование и верификацию на реальных данных.

РЭФЕРАТ

Дыпломная работа – 63 старонкі, 12 малюнкаў, 1 табліца, 15 крыніц.

ГЕАІНФАРМАЦЫЙНЫЯ СІСТЭМЫ, АГРАХІМІЯ, DRUPAL 10, PHP 8.4, MYSQL, BOOST.GEOMETRY, LEAFLETJS, OPENSTREETMAP, DOCKER, GIT, WATCH API.

Практычная значнасць працы заключаецца ў распрацоўцы і ўкараненні модуля геапрасторавага аналізу ў рамках банка даных для станцый хімізацыі. Рашэнне накіравана на аўтаматызацыю захоўвання, адлюстравання і апрацоўкі прасторавых і часавых звестак пра элементарныя ўчасткі сельскагаспадарчых земляў.

Мэтай працы з'яўляецца распрацоўка модуля геапрасторавага аналізу, які забяспечвае фіксацыю, візуалізацыю і параўнанне геаметрычных даных участкаў у розныя туры абследавання, а таксама падтрымку сувязяў паміж участкам і іх папярэднікамі.

Задачы:

1. Прааналізаваць дзейнасць станцый хімізацыі і вызначыць ключавыя патрабаванні да модуля.
2. Спраектаваць архітэктuru праграмага рашэння з улікам працы з геаданымі.
3. Рэалізаваць модуль на платформе Drupal 10 з выкарыстаннем PHP і MySQL.
4. Забяспечыць візуалізацыю участкаў на карце з прымяненнем LeafletJS і OpenStreetMap.
5. Рэалізаваць алгарытм вызначэння перасячэнняў участкаў паміж турамі на аснове Boost.Geometry і метаду Вейлера-Азертана.
6. Забяспечыць магчымасць захоўвання інфармацыі пра спадчынныя сувязі участкаў.
7. Правесці тэставанне і верыфікацыю на рэальных даных.

ABSTRACT

Thesis – 63 pages, 12 figures, 1 table, 15 references.

GEOGRAPHIC INFORMATION SYSTEMS, AGROCHEMISTRY, DRUPAL 10, PHP 8.4, MYSQL, BOOST.GEOMETRY, LEAFLETJS, OPENSTREETMAP, DOCKER, GIT, BATCH API.

The practical significance of this work lies in the development and implementation of a geospatial analysis module within a data bank for agrochemical stations. The solution is aimed at automating the storage, visualization, and processing of spatio-temporal information on elementary land plots used in agriculture.

The goal of the work is to develop a geospatial analysis module that provides fixation, visualization, and comparison of geometric data of land plots across different survey rounds, as well as maintaining links between plots and their predecessors.

Objectives:

1. Analyze the activities of agrochemical stations and define key requirements for the module.
2. Design the architecture of the software solution, considering the specifics of working with geodata.
3. Implement the module on the Drupal 10 platform using PHP and MySQL.
4. Provide visualization of land plots on a map using LeafletJS and OpenStreetMap.
5. Implement an algorithm for identifying intersections of plots between survey rounds based on Boost.Geometry and the Weiler-Atherton method.
6. Enable the storage of information about the inheritance relationships between land plots.
7. Perform testing and verification on real-world data.

ОГЛАВЛЕНИЕ

ВВЕДЕНИЕ.....	7
Глава 1 Современное состояние информатизации в агропромышленном комплексе. «СОД Агро 2.0».....	9
1.1 Информационные технологии в агрохимии.....	9
1.2 «СОД Агро 2.0»	11
1.3 Drupal 10 как основа для программного комплекса «СОД Агро 2.0»	12
Глава 2 Архитектура и компоненты программного комплекса	15
2.1 Основные модули и их функциональное назначение	15
2.2 Организация доступа, безопасности и разграничения прав	18
2.3 Выбор технологий: обоснование использования Drupal 10 и MySQL	20
2.4 Импорт, хранение и обработка данных: использование Batch API и оптимизация ресурсоемких операций.....	21
Глава 3 Пространственно-временная модель агрохимического мониторинга.....	23
3.1 Понятие тура, элементарного и рабочего участка.....	23
3.2 Структура и иерархия объектов в системе	24
3.3 Проблематика идентификации участков во времени.....	27
3.4 Работа с пространственными данными: формат SHP и геометрическое описание участков.....	30
Глава 4 Разработка модуля геопространственного анализа	32
4.1 Цель и задачи модуля	32
4.2 Структура модуля и алгоритм работы	33
4.2.1 Архитектура модуля	33
4.2.2 Алгоритм Вейлера-Азерттона: теория и применение.....	34
4.2.3 Сравнительный анализ алгоритма Сазерленда-Ходгмана.....	43
4.2.4 Расчёт степени вхождения и родства по площади	45
4.3 Интерфейс взаимодействия с модулем и визуализация данных.....	47
4.4 Методология и ведение разработки	48

Глава 5 Результаты и перспективы использования модуля.....	53
5.1 Результаты разработки	53
5.2 Возможные нововведения в модуль и перспективы развития	58
Заключение	61
СПИСОК ЛИТЕРАТУРЫ.....	64

ВВЕДЕНИЕ

Областные проектно-изыскательские станции химизации (ОПИСХ) выполняют важнейшую функцию в системе агрохимического обеспечения аграрного сектора в Республике Беларусь. Эти организации занимаются проведением анализа состояния почв, мониторингом агрохимических показателей, прогнозированием потребностей в удобрениях и известковании, а также оценкой экономической обоснованности агрохимического обслуживания сельскохозяйственных угодий.

Одной из приоритетных задач ОПИСХ является ведение агрохимического мониторинга земель, включающего сбор, хранение и систематизацию информации о состоянии почв. Это позволяет своевременно отслеживать изменения в показателях плодородия, содержание элементов питания, наличие радионуклидов в почвах, а также определения факторов, влияющих на продуктивность сельскохозяйственного производства. На основе этих данных формируются научно обоснованные рекомендации для агропредприятий, хозяйств и государственных структур.

С увеличением объёмов обрабатываемой информации и возрастающими требованиями к точности и оперативности анализа возникает необходимость в создании современного программного комплекса, способного автоматизировать ключевые процессы сбора, обработки и визуализации данных. Целью настоящей дипломной работы является разработка компонента этого комплекса – **модуля геопространственного анализа**, который станет частью программного комплекса под рабочим названием «*СОД Агро 2.0*».

Программный модуль геопространственного анализа должен обеспечивать приём, хранение и обработку пространственных данных, привязанных ко времени обследования. Он будет выполнять визуализацию обследуемых территорий на карте, поддерживать манипулирование геометрическими объектами и обеспечивать структурированное хранение информации о взаимосвязях между различными обследованными участками в

разные периоды. Всё это необходимо для построения целостной цифровой модели, отражающей пространственно-временные изменения на уровне земельных массивов.

Настоящая работа направлена на проектирование и реализацию модуля геопро пространственного анализа в рамках процесса информатизации республиканских станций химизации.

Глава 1 Современное состояние информатизации в агропромышленном комплексе. «СОД Агро 2.0».

1.1 Информационные технологии в агрохимии

Современное сельское хозяйство немислимо без применения информационных технологий. Цифровизация агропромышленного комплекса охватывает широкий спектр задач: от автоматизированного учета ресурсов и планирования сельскохозяйственных работ до систем точного земледелия и мониторинга состояния почв и растений. Внедрение информационных решений позволяет повысить эффективность производства, минимизировать издержки, обеспечить устойчивое использование природных ресурсов и повысить качество управленческих решений.

Одним из ключевых направлений в процессе цифровой трансформации является создание информационных систем, интегрированных с геоданными. Такие системы позволяют привязывать сведения о сельскохозяйственных объектах к конкретной местности, что особенно важно при мониторинге и анализе пространственно распределённых данных. В том числе это актуально для агрохимических служб, работа которых требует точной фиксации координат обследуемых территорий, сохранения информации о химическом составе почв и отслеживания изменений во времени.

В Республике Беларусь, как и во многих других странах, активно реализуются программы по информатизации аграрной отрасли. Одним из элементов этой работы является развитие автоматизированных банков данных, предназначенных для хранения, систематизации и анализа информации, получаемой в ходе агрохимических обследований и производственной деятельности.

Станции химизации, такие как ОПИСХ, на протяжении десятилетий формируют массивы данных по агрохимическому состоянию почв. Эти данные включают информацию о содержании макро- и микроэлементов, кислотности,

степени загрязнения, в том числе радионуклидами, а также рекомендации по внесению удобрений и известкованию.

Несмотря на высокую значимость этих сведений, методы их учета и анализа зачастую остаются недостаточно автоматизированными. Основная часть информации ведется в виде таблиц, отчетов и картографических материалов, подготовленных вручную или с применением простых графических редакторов. Это затрудняет обработку больших объемов данных, делает невозможной оперативную аналитику и повышает риск ошибок.

Сложности возникают и при отслеживании изменений обследованных территорий во времени. Без централизованного и формализованного хранения пространственных данных становится трудно фиксировать перемещения участков, изменения их границ, а также анализировать повторные обследования. Отдельную проблему представляет организация и поддержка связей между разными периодами наблюдений, особенно в тех случаях, когда территории подвергались перераспределению или укрупнению.

Все эти факторы свидетельствуют о необходимости внедрения современных решений в области геоинформационных технологий и баз данных. Разработка специализированного модуля, способного работать с пространственными данными и обеспечивать поддержку структурированной информации, является актуальной задачей для агрохимических служб.

1.2 «СОД Агро 2.0»

Ранее на станциях химизации использовали ПО, позволяющее производить необходимые калькуляции и формировать выходную документацию. Это позволяло автоматизировать процесс расчетов необходимых данных и уменьшить вероятность ошибки. Однако, с течением времени требования и задачи изменяются. Данные так или иначе приходится перепроверять, для групповых смет требуется пересмотр и сопоставление сгенерированной выходной документации. Различные требования разных отделов могут приводить к необходимости пересмотра уже сформированных документов. Нынешняя система уже не является адаптивной и масштабируемой. На данный момент требуется высокий уровень согласованности между различными отделами, электронное хранение документации, и, помимо этого, нужен совершенно новый функционал, такой как отслеживание истории участков относительно географических изменений их форм, площадей и расположения. Ввиду данных причин ведутся активная разработка и тестирование программного комплекса «СОД Агро 2.0» -- в тестировании активно участвуют конечные пользователи их различных отделов и областей Республики Беларусь, что позволяет оперативно разрабатывать необходимый функционал и делать комплекс максимально удовлетворяющим потребности конечного пользователя. На данный момент комплект уже имеет большое количество функционала и модулей – ввод входных данных в удобном формате, огромная вариативность импорта данных из формируемых предыдущим ПО документов и файлов, формирование финансово-сметной отчетности, агрохимических паспортов почв. Помимо этого, в системе уже присутствует удобный графический интерфейс, надежная система защиты данных, как от потери, так и от несанкционированного доступа к данным, четкое распределение ролей и доступа к отдельному функционалу программного комплекса для различных категорий сотрудников станций химизации. На момент написания система, все еще находясь на стадии активной разработки и тестирования, уже

насчитывает более полумиллиона объектов.

1.3 Drupal 10 как основа для программного комплекса «СОД Агро 2.0»

Разработка программного комплекса «СОД Агро 2.0» велась на базе современной версии Drupal – а именно, Drupal 10. Это не просто популярная CMS, как её часто представляют, а полноценный каркас для построения сложных и разветвлённых web-систем. В последние годы Drupal перерос рамки систем «для публикации статей» и стал мощным инструментом для прикладных решений, в том числе отраслевых – таких как агрохимические информационные системы.

Drupal 10 вообрал в себя возможности PHP-фреймворков нового поколения, включая Symfony, и сочетает гибкость архитектуры с широкой модульной системой. В случае «СОД Агро 2.0» эта платформа позволила реализовать специфический функционал без необходимости создавать базовые вещи с нуля – такие как маршруты, формы, работа с пользователями, разграничение прав доступа и многое другое.

Важно понимать, что Drupal работает не только с «контентом» в привычном понимании. Его сущностная модель построена на концепции абстрактных объектов (entities), которые можно настраивать под любую предметную область. В нашем случае – это рабочие участки, хозяйства, данные агрохимического анализа, элементы известкования и пр. Всё это реализуется через кастомные типы сущностей, что позволяет гибко описывать поля, связи между объектами, форматы ввода и отображения.

Программный комплекс «СОД Агро 2.0» активно использует эту гибкость. Например, агрохимические показатели по участкам формируются как поля у нод, а терминология (та же классификация типов угодий или почв) – через таксономические словари. Но за пределами пользовательского интерфейса стоит немало логики. Drupal позволяет создавать собственные модули, которые

внедряются в ядро системы на равных правах. Именно через них реализованы алгоритмы расчёта доз извести, анализ изменений между турами обследования, генерация отчётных PDF-документов и многое другое.

Ещё одно важное преимущество – поддержка миграций и импорта данных. Сельхозданные часто хранятся в разрозненных источниках: это могут быть старые Excel-таблицы, CSV-файлы или даже сканы. Drupal предоставляет инструменты для загрузки этих данных в структурированном виде, а также возможности валидации и дальнейшей автоматической обработки.

С точки зрения сопровождения, Drupal тоже даёт плюсы. Все конфигурации можно хранить в YAML-файлах, что позволяет переносить структуру между средами – от локальной разработки до боевого сервера. Это удобно как при работе в команде, так и для будущей поддержки.

Кроме того, в экосистеме Drupal существует множество готовых решений, которые можно подключать без существенных переделок. Это касается, например, работы с картами – модуль Leaflet прекрасно справляется с визуализацией участков на базе координат. Также присутствуют API-инструменты – можно настраивать REST-интерфейсы или использовать JSON:API, если потребуется интеграция с внешними системами.

Что немаловажно – система хорошо масштабируется. При необходимости её можно разворачивать на кластере, подключать к Redis или Memcached для кэширования, использовать Composer для управления зависимостями, и применять современные CI/CD-практики. Таким образом, даже если в будущем проект вырастет в более крупную платформу (например, на уровне республиканской системы агромониторинга), техническая основа не потребует полной переделки.

Drupal 10, конечно, требует определённого входного порога. Это не конструктор, где всё делается в пару кликов. Но именно это и делает его подходящим для серьёзных задач. Возможность полного контроля над системой, чёткое разделение логики, расширяемость – всё это делает его рациональным

выбором для комплексной отраслевой разработки.

В целом, выбор Drupal 10 как технологической базы оказался оправданным: и с точки зрения архитектуры, и с позиции темпов разработки, и с учётом сопровождения. Система показала себя как надёжная основа, позволяющая реализовать как текущие, так и потенциальные задачи «СОД Агро 2.0».

Глава 2 Архитектура и компоненты программного комплекса

2.1 Основные модули и их функциональное назначение

В рамках процесса цифровой трансформации агрохимического обслуживания в Республике Беларусь активно разрабатывается и проходит апробацию программный комплекс с рабочим названием «СОД Агро 2.0». Система представляет собой банк данных, предназначенный для хранения, структурирования и обработки информации, получаемой в ходе агрохимических обследований сельскохозяйственных угодий.

Основной задачей системы является обеспечение централизованного доступа к актуальным и историческим агрохимическим данным, формирование отчётности, а также организация внутренней аналитической и административной работы специалистов. «СОД Агро 2.0» служит корпоративным инструментом для областных проектно-изыскательских станций химизации сельского хозяйства (ОПИСХ) [1], а также других уполномоченных структур.

Система позволяет вводить и хранить данные, относящиеся к различным уровням административного деления – от хозяйств до районов и областей. Важной архитектурной особенностью является поддержка иерархии пользователей: в зависимости от должности и полномочий сотрудники получают доступ к соответствующим разделам и функциям системы.

Также в рамках проекта реализуются механизмы формирования отчётности, ведения истории изменений и контроля полноты введённых сведений. Таким образом, создаваемая система выступает как инфраструктурная платформа, способствующая информатизации агрохимического мониторинга и административной деятельности в данной области.

Программный комплекс «СОД Агро 2.0» представляет собой комплексный банк данных веб-формата, разработанный для обеспечения централизованного хранения, обработки и анализа агрохимической информации. В рамках

платформы реализован ряд модулей, каждый из которых выполняет строго определённую функциональную задачу в соответствии с требованиями проектных и эксплуатационных подразделений ОПИСХ.

Одним из ключевых компонентов является **система учёта пользователей и ролей**, обеспечивающая разграничение прав доступа. Благодаря иерархической структуре ролей сотрудники могут взаимодействовать только с теми модулями и данными, которые соответствуют их компетенциям и уровню полномочий. Это повышает безопасность и исключает несанкционированное изменение критически важной информации.

Фундаментальную роль играет **модуль учета земельных участков** – на данный момент в систему внесены сведения о более чем 200 000 участках. Для каждого объекта хранится структурированная информация о координатах, административной принадлежности, а также результатах агрохимических обследований. Данные могут быть добавлены вручную или импортированы из различных файловых форматов.

Система поддерживает **импорт данных о хозяйствах**, их последующую регистрацию и редактирование. Загружаемые сведения включают агрохимические характеристики участков, на которых проводились обследования. Формы ввода и редактирования спроектированы с учётом высокой степени автоматизации – реализована поддержка импорта из файлов .csv, .dbf, а также механизмы вставки информации по столбцам из буфера обмена. Это позволяет оперативно загружать большие объёмы информации и снижает трудозатраты специалистов.

Модуль ведомостей кодирования позволяет формировать стандартизированные записи по участкам: включаются данные о насыщенности радионуклидами, содержании гумуса, фосфора (P_2O_5), калия (K_2O) и других показателях. Записи создаются на основе методических рекомендаций и стандартизированы по формам, используемым в ОПИСХ. Все данные сохраняются в банк данных с возможностью последующего редактирования,

сопоставления и анализа.

На основе внесённой информации функционирует **модуль расчёта известкования**, который позволяет вычислить необходимое количество извести для конкретных участков в соответствии с методиками, применяемыми в профильных организациях. Аналогичным образом реализован **модуль расчёта потребности в минеральных удобрениях**, использующий установленные агрохимические алгоритмы и нормативы. По результатам расчётов формируется документация, пригодная для использования в отчетных и плановых целях.

Ещё одной важной частью платформы является **модуль агрохимпаспортов**, обеспечивающий автоматическое формирование отчётных документов, отражающих текущее состояние и изменения состава почв. Сформированные агрохимпаспорта могут быть сгруппированы по административной принадлежности, хозяйствам и другим параметрам, что упрощает последующую аналитическую работу.

Дополнительно реализован **интерфейс графического просмотра данных по хозяйству**, обеспечивающий визуализацию обследованных участков и их агрохимических характеристик. Это позволяет специалистам получать наглядное представление о распределении показателей по территории.

Техническая реализация модулей базируется на устойчивой архитектуре, построенной на платформе Drupal 10. Используются настраиваемые шаблоны объектов (участков, хозяйств, ведомостей и др.), что обеспечивает логически выстроенные взаимосвязи между сущностями, их надёжное хранение и консистентный доступ. Система демонстрирует стабильную работу с большими объёмами данных, что критически важно в условиях масштабной агрохимической деятельности.

На текущем этапе программный комплекс «СОД Агро 2.0» уже обладает широким спектром реализованного функционала, включая механизмы ведения банка данных, управления пользовательским доступом, расчёта агрохимических показателей, импорта и структурирования данных, а также формирования

отчетной документации. Однако система продолжает активно развиваться и дорабатываться с учетом возникающих задач и требований, формируемых в процессе эксплуатации, в том числе со стороны ОПИСХ.

Платформа спроектирована как модульная, что позволяет органично расширять её функциональные возможности без нарушения архитектурной целостности. Одним из таких направлений расширения является разработка **модуля геопространственного анализа**, который обеспечивает фиксацию и визуализацию геометрии обследуемых объектов, а также позволяет отслеживать изменения на различных этапах обследования.

В рамках дальнейшего развития предусматривается внедрение дополнительных специализированных решений. В частности, **разработка модуля оценки экономической обоснованности мероприятий по известкованию и применению удобрений** на различных участках. Такой функционал позволит опираться не только на агрохимические показатели, но и учитывать экономические параметры – потенциальную урожайность, стоимость материалов, окупаемость и пр. Это даст возможность более эффективно планировать мероприятия в рамках агрохимического сопровождения и поддержки хозяйств.

Кроме того, система будет адаптироваться под возможные изменения в нормативных и методических документах, используемых профильными организациями, что обеспечит её актуальность и соответствие действующим требованиям.

2.2 Организация доступа, безопасности и разграничения прав

Программный комплекс «СОД Агро 2.0» предусматривает строгую организацию доступа к данным и функциональным модулям, обеспечивая безопасность и целостность информации.

Система реализует иерархическую модель ролей, соответствующую структуре ОПИСХ. Каждому пользователю присваивается определённая роль,

определяющая его права и доступ к функционалу:

1. **Почвовед:** ввод и редактирование данных по почвенным характеристикам.
2. **Лаборант:** обработка лабораторных анализов и внесение результатов.
3. **Агрохимик:** анализ агрохимических данных и формирование рекомендаций.
4. **Сотрудник отдела ГИС:** работа с геопространственными данными, без доступа к агрохимической информации.

Незарегистрированные пользователи не имеют доступа к системе. Каждый пользователь связан с определённой областью, что ограничивает его действия в рамках соответствующего региона.

Для обеспечения безопасности данных в системе реализованы следующие механизмы:

1. **Хеширование паролей с использованием соли:** Drupal 10 применяет функцию `password_hash()` с уникальной солью для каждого пользователя, что обеспечивает защиту от атак методом перебора и использования радужных таблиц [2].
2. **Защита от CSRF-атак:** все запросы на изменение данных подписываются CSRF-токеном, который проверяется на стороне сервера, предотвращая несанкционированные действия [3].
3. **Ограничение доступа к файлам через .htaccess:** система использует файл `.htaccess` для ограничения прямого доступа к конфиденциальным файлам, обеспечивая дополнительный уровень защиты. Файл `.htaccess` – это конфигурационный файл веб-сервера Apache. В нём можно настроить перенаправления, поменять параметры PHP, установить права доступа к файлам и каталогам, задать собственные страницы ошибок и многое другое [4].

В настоящее время ведётся разработка механизма, предотвращающего одновременное редактирование одних и тех же данных несколькими

пользователями. Это позволит избежать конфликтов и потери данных при параллельной работе.

Таким образом, «СОД Агро 2.0» обеспечивает надёжную систему управления доступом и защиту данных, соответствующую современным требованиям безопасности.

2.3 Выбор технологий: обоснование использования Drupal 10 и MySQL

Разработка современных веб-ориентированных систем требует выбора технологического стека, обеспечивающего масштабируемость, гибкость, безопасность и удобство сопровождения. В рамках построения программного комплекса агрохимического мониторинга в качестве базовых решений были выбраны CMS Drupal 10 и СУБД MySQL.

Drupal 10 представляет собой мощную, гибкую и модульную систему управления контентом, которая активно применяется для создания масштабируемых информационных решений, включая государственные и корпоративные порталы, банки данных и экспертные платформы. Среди ключевых преимуществ платформы можно выделить:

1. **Модульность и расширяемость:** ядро Drupal предоставляет обширные возможности для расширения функционала посредством модулей, в том числе пользовательских, с сохранением целостности архитектуры.

2. **Система управления доступом:** Drupal реализует сложную систему разграничения прав и ролей, что критично для программных комплексов с различными уровнями пользователей.

3. **Поддержка сложных связей между объектами:** возможность построения гибких схем данных и хранения метаданных в виде сущностей и полей.

4. **Совместимость с современными стандартами безопасности:** встроенная защита от XSS, CSRF, SQL-инъекций и других уязвимостей.

5. Большое сообщество и поддержка: активное сообщество и регулярные обновления делают Drupal одним из наиболее надёжных решений в сфере CMS.

С точки зрения архитектуры и возможности интеграции с внешними сервисами и системами, Drupal 10 предоставляет достаточную свободу для реализации как стандартных, так и специализированных решений, что делает его подходящим выбором для целей данной системы.

На практике выбор между реляционными системами управления базами данных, такими как PostgreSQL и MySQL, зависит от специфики проекта. В контексте Drupal 10 оба варианта поддерживаются официально, однако MySQL имеет ряд преимуществ с точки зрения производительности и совместимости.

Согласно результатам сравнительного анализа производительности различных СУБД в среде Drupal 10, представленных в исследовании Белорусского государственного университета информатики и радиоэлектроники [5], MySQL продемонстрировала более высокие показатели скорости обработки запросов и масштабируемости при типичных операциях ввода-вывода и выборки, чем PostgreSQL. Это особенно критично при работе с большими объёмами данных и интенсивной нагрузке на систему.

Дополнительно, учитывая широкую распространённость MySQL, большое число специалистов, документации и инструментов мониторинга, выбор данной СУБД позволяет снизить потенциальные риски при сопровождении и масштабировании системы.

Несмотря на тот факт, что PostgreSQL имеет большую популярность в области хранения географических данных, в рамках данного проекта MySQL более предпочтителен по другим параметрам, важным для системы в целом.

2.4 Импорт, хранение и обработка данных: использование Batch API и оптимизация ресурсоемких операций

Работа с агрохимическими данными предполагает обработку

значительных массивов информации, включая импорт сведений о хозяйствах и участках, массовое заполнение ведомостей, формирование отчётности, а также выполнение расчётов, основанных на специализированных методиках. В условиях высокой нагрузки на сервер и ограничений по времени выполнения PHP-скриптов (особенно при импорте файлов большого объёма) возникает необходимость использования средств, позволяющих организовать асинхронную и устойчивую обработку данных.

Для решения данной задачи в системе используется **Batch API**, входящий в состав ядра Drupal. Этот механизм предназначен для разбиения ресурсоёмких операций на небольшие итеративные шаги с сохранением промежуточного состояния, что позволяет:

избегать превышения лимитов времени выполнения скриптов;

отслеживать прогресс выполнения операции и уведомлять пользователя о статусе процесса;

обеспечивать устойчивость при временных сбоях и возможность повторного запуска задачи с сохранением данных.

В текущей реализации Batch API уже применяется в следующих компонентах системы:

Модули импорта данных: при загрузке ведомостей кодирования, хозяйственных и участковых данных из CSV, DBF и других форматов. Обработка каждой строки данных производится последовательно, с минимальным потреблением ресурсов;

Механизмы формирования агрохимических паспортов и другой отчётной документации: построение документов осуществляется поэтапно, с разбивкой по логическим блокам и хозяйствам;

Алгоритмы расчёта доз известкования и рекомендаций по удобрениям: модульные расчёты производятся через очередь заданий, что позволяет масштабировать вычисления без задержек пользовательского интерфейса.

Глава 3 Пространственно-временная модель агрохимического мониторинга

3.1 Понятие тура, элементарного и рабочего участка

В контексте агрохимического мониторинга и деятельности областных проектно-изыскательских станций химизации сельского хозяйства (ОПИСХ) ключевыми понятиями при работе с пространственными и агрохимическими данными являются *тур*, *элементарный участок* и *рабочий участок*.

Элементарный участок представляет собой минимально возможную, *неделимую с точки зрения агрохимического анализа* единицу обследования. Он характеризуется тем, что любые агрохимические показатели (например, содержание гумуса, подвижных форм фосфора и калия, кислотность, наличие радионуклидов и пр.) рассматриваются в пределах участка как единые, даже если в пределах его площади наблюдаются незначительные отклонения. Таким образом, наличие специфического элемента даже в незначительном объёме фиксируется как характеристика всего элементарного участка.

Рабочий участок является более крупной агрегацией – это группа элементарных участков, которые обладают сходными характеристиками, в частности: механическим составом почвы, содержанием питательных элементов, уровнем кислотности и др. Он служит своего рода методологической абстракцией, упрощающей расчёты и аналитическую работу, особенно при формировании сводной информации. Однако со временем границы рабочих участков и состав входящих в них элементарных участков могут изменяться в связи с новыми данными или пересмотром методологии.

Тур – это период между последовательными обследованиями одного и того же участка почвы. В рамках одного тура сотрудник ОПИСХ непосредственно выезжает на сельскохозяйственные угодья, осуществляет сбор проб, визуальное обследование, а также определяет и размечает границы элементарных участков. После этого данные поступают в соответствующие

отделы, включая отдел геоинформационных систем, где осуществляется пространственная фиксация границ, и лаборатории, в которых выполняются химические и физические анализы почвы.

Тур является важнейшим понятием во временной модели мониторинга, поскольку позволяет отслеживать изменения агрохимических характеристик участков в динамике. Как правило, стандартный интервал между турами составляет **четыре года**, однако в реальности этот период может варьироваться. Причинами отклонения от стандартного цикла могут быть особенности обследования различных административных единиц (районов, областей), внутренние графики работы станции, погодные условия, необходимость внепланового анализа и другие факторы.

Понимание структуры и логики этих понятий – элементарного участка, рабочего участка и тура – лежит в основе корректного проектирования и реализации информационной модели агрохимического мониторинга, которая обеспечивает надёжную фиксацию, анализ и отслеживание изменений состояния почвенных ресурсов.

3.2 Структура и иерархия объектов в системе

Проектирование и реализация структур хранения данных в рамках разрабатываемого программного комплекса агрохимического мониторинга выполняется на платформе **Drupal 10**, что во многом определяет архитектурную основу и принципы взаимодействия всех компонентов системы. Drupal – это мощная и гибкая система управления контентом (CMS), построенная на языке PHP и обладающая выраженной объектно-ориентированной архитектурой, что делает её особенно подходящей для создания сложных масштабируемых микросервисных систем, ориентированных на обработку большого количества структурированных данных.

В системе управления контентом Drupal **нода (node)** представляет собой базовую программную сущность, описывающую единицу информации,

размещённую на сайте. Ноды являются основой для создания **страниц определённых типов**, каждый из которых может быть заранее определён и структурирован посредством административного интерфейса. Типы нод (например, «Элементарный участок», «Хозяйство», «Отчёт») определяют набор доступных полей, формат отображения и поведение на уровне представления.

С архитектурной точки зрения, ноды реализованы как экземпляры одной из ключевых сущностей Drupal – **контентных сущностей (content entities)**. Наряду с нодами, к этому классу относятся также **термины таксономии, пользователи, комментарии, медиа-объекты** и другие объекты, управляемые системой. Все сущности построены на общем API и поддерживают механизмы расширения, хранения произвольных данных, а также связывания между собой через ссылочные поля.

Такой подход позволяет придерживаться **объектно-ориентированной модели**, в которой каждая нода представляет собой изолированный и расширяемый объект с собственным набором данных, связей и поведения. Использование встроенных API системы (Field API, Entity API, Form API и др.) значительно упрощает разработку, обеспечивая модульность, масштабируемость и повторное использование компонентов.

Таксономия – это механизм классификации, позволяющий создавать иерархические и неиерархические словари терминов. Она применяется для структурирования информации, категоризации контента и реализации связей. Например, «Тур» может быть представлен как термин в словаре «Периоды обследования», а «Район» или «Область» – как термины в соответствующих географических таксономиях.

Параграфы – это особый тип сущностей, позволяющий создавать гибкие группы полей, которые могут повторяться, структурироваться и переиспользоваться внутри одной ноды. Это особенно полезно при необходимости хранения вложенных, сложных данных (например, динамически повторяющихся блоков по результатам анализа на разные элементы).

В рамках разрабатываемого модуля и всей СОД «АГРО» используется структурная модель, в основе которой лежит концепция объектно-ориентированного представления агрохимических данных.

Элементарный участок в системе реализован как *нода*, содержащая полный набор агрохимических показателей: содержание гумуса, кислотность, рН, подвижные формы фосфора (P_2O_5), калия (K_2O), кальция, магния, а также наличие радионуклидов и других компонентов. Все эти данные хранятся в отдельных числовых полях, привязанных к соответствующему типу содержимого. В том числе, в ноде элементарного участка хранятся:

Ссылочное поле на **рабочий участок**, к которому он принадлежит,

Ссылочное поле на **термин таксономии тура**, указывающее, к какому периоду обследования относится данный участок,

Ссылочное поле на **хозяйство**, которое представлено как отдельная нода с собственными полями (код хозяйства, УНП, район, область и пр.).

Ссылочные, числовые и текстовые поля для хранения информации о содержании элементов и классификации почвы.

Рабочий участок также является нодой и выступает в роли агрегирующего объекта. Он содержит информацию о типе почвы, а также связи с элементарными участками.

Хозяйство, как самостоятельная сущность, предоставляет возможность группировать участки и управлять информацией на уровне административно-хозяйственных единиц. Благодаря ссылкам на ноды хозяйств, становится возможным фильтрация, построение отчётности и аналитика на уровне как отдельного хозяйства, так и объединённых территорий.

Тур – реализован в виде термина таксономии, позволяющего не только классифицировать данные по периодам обследования, но и упрощать временной анализ изменений состава почв. Использование таксономии даёт возможность централизованно управлять списками доступных туров и их структурой.

Данная модель, реализованная на архитектуре Drupal, обладает целым

рядом преимуществ:

1. **Независимость объектов** – каждый элементарный участок является самостоятельной сущностью, которую можно отслеживать, редактировать, связывать или анализировать отдельно.

2. **Связность и масштабируемость** – с помощью ссылочных полей и таксономии обеспечивается строгая логика связей между объектами, что облегчает миграцию данных, их обработку и построение отчетности.

3. **Инкапсуляция логики** – вся бизнес-логика, связанная с отображением, расчетами, доступом и валидацией данных, может быть инкапсулирована в модулях, что соответствует принципам объектно-ориентированного подхода.

4. **Расширяемость** – при изменении требований (например, добавление новых показателей или параметров анализа), система позволяет легко добавить новые поля, связанные сущности и методы обработки без необходимости глубокой переработки архитектуры.

5. **Поддержка безопасности и контроля доступа** – каждая сущность управляется системой прав и ролей Drupal, обеспечивая строгий контроль за доступом к данным, включая разграничение по должностям и регионам.

Данная модель позволяет получить удобный интерфейс для программной реализации необходимого функционала через сторонние или самостоятельно реализуемые модули и плагины, а объектно-ориентированный подход позволяет реализовывать их масштабируемыми и безопасными.

3.3 Проблематика идентификации участков во времени

Одной из наиболее острых и актуальных задач в агрохимическом мониторинге является **отслеживание идентичности и преемственности элементарных участков** в рамках различных туров обследования. На текущем этапе развития системы данная проблема стоит особенно остро и требует специального внимания, как со стороны архитектуры базы данных, так и со

стороны логики обработки пространственно-временной информации.

Первой ключевой особенностью агрохимического мониторинга является **нефиксированность длительности тура**. Несмотря на то, что официально один тур предполагает прохождение полного обследования территории каждые четыре года, в реальности это значение варьируется. Периоды могут составлять **три года, два с половиной или даже больше четырех лет**, в зависимости от ряда факторов:

- 1) логистических ограничений;
- 2) изменений в расписании обследований;
- 3) приоритетов по отдельным регионам;
- 4) технических или кадровых сложностей.

Таким образом, **тур не является объективной временной меткой**, а скорее представляет собой условную фазу обследования, привязанную к конкретной географической и административной области и зависящую от оперативного графика работы сотрудников. Это уже накладывает определённые трудности на формирование чёткой временной модели.

Наиболее сложная часть проблемы – **потеря преемственности между элементарными участками** от одного тура к другому. С каждым новым обследованием:

- 1) **меняются границы** участков;
- 2) **изменяется площадь** (например, участок может быть объединён с соседними и стать больше);
- 3) **изменяется положение** (участок может быть переразмечен, смещён, включать или исключать новые зоны);
- 4) **изменяется нумерация**, и она не сохраняет никакой логической связи с предыдущими турами.

То есть участок с определенным номером в первом туре, с определенной площадью, содержащий определённый набор агрохимических характеристик, может в следующем туре превратиться в участок под другим номером, с

совершенно другой площадью, и не иметь **никакой формальной связи** с предыдущим. Такая трансформация может происходить **автоматически или в результате ручной корректировки**, и не **отслеживается системой** на уровне идентификаторов или логических связей.

На данный момент **не существует механизмов уникальной идентификации** элементарных участков между турами. Используемые в системе номера являются **локальными идентификаторами**, которые актуальны только в рамках одного тура. Как результат:

- 1) невозможно однозначно сопоставить участки между турами;
- 2) невозможно автоматически построить временные ряды изменений агрохимических характеристик на одной и той же территории;
- 3) невозможно оценивать тренды и динамику на постоянной территории, без привлечения дополнительных источников информации.

В случае необходимости анализа изменений, **вся нагрузка по сопоставлению ложится на отдел ГИС**, сотрудники которого должны вручную анализировать геометрические и пространственные параметры участков с использованием ГИС-программ (например, QGIS, MapInfo, ArcGIS и т. д.). Это делает процесс **трудозатратным, чувствительным к человеческому фактору**, и не допускающим автоматизации.

Дополнительным осложняющим фактором является **высокая вероятность ошибок ввода и непредсказуемость нумерации**, особенно при больших объёмах данных (в системе уже содержатся сведения по более чем 200 000 участках, при условии того что система на момент написания находится в разработке и тестирование отдельных модулей было начато ОПИСХами совсем недавно). Человеческий фактор в таких условиях неизбежно приводит к нарушению целостности и преемственности данных.

3.4 Работа с пространственными данными: формат SHP и геометрическое описание участков

Для эффективного управления агрохимическими данными и пространственной информацией в нашей системе используется формат **Shapefile (SHP)** – один из наиболее распространённых векторных форматов в геоинформационных системах (ГИС). Этот формат разработан компанией Esri и широко применяется для хранения и обмена географическими данными между различными программными продуктами [7].

Формат Shapefile представляет собой набор файлов с одинаковым именем, но разными расширениями, каждый из которых выполняет определённую функцию:

- 1) **.shp** – основной файл, содержащий геометрические данные объектов (точки, линии, полигоны);
- 2) **.shx** – индексный файл, обеспечивающий быстрый доступ к геометрическим данным в **.shp**;
- 3) **.dbf** – таблица атрибутов в формате dBase, содержащая описательные данные для каждого объекта.

Дополнительно могут присутствовать файлы **.prj** (содержащий информацию о системе координат), **.sbn** и **.sbx** (пространственные индексы), а также другие вспомогательные файлы.

В контексте агрохимического мониторинга, каждый **элементарный участок** представляется в виде **полигона**, описанного последовательностью координат (x, y), определяющих его границы. Эти полигоны хранятся в файле **.shp** и визуализируются на карте как замкнутые области.

Атрибутивные данные, такие как идентификатор участка, содержание химических элементов, тип почвы и другие характеристики, хранятся в соответствующих записях файла **.dbf**. Каждая запись в **.dbf** соответствует одному геометрическому объекту в **.shp**, что обеспечивает однозначную связь между пространственными и описательными данными [8].

В нашей системе Shapefile-файлы используются для:

1. Импорта данных о геометрии участков, полученных из внешних источников или в результате полевых обследований.
2. Хранения и отображения пространственной информации о участках в интерфейсе пользователя.
3. Обеспечения возможности пространственного анализа и визуализации данных в ГИС-приложениях.

Благодаря широкому распространению и поддержке в различных программных продуктах, использование формата Shapefile обеспечивает совместимость и удобство интеграции с другими системами и инструментами, используемыми в агрохимическом мониторинге.

Глава 4 Разработка модуля геопространственного анализа

4.1 Цель и задачи модуля

Целью разработки модуля геопространственного анализа является создание инструмента, позволяющего проводить автоматизированное сопоставление **элементарных участков** между различными турами агрохимического обследования на основе их геометрического положения. Модуль призван решить одну из ключевых проблем агрохимического мониторинга – отсутствие системы идентификации и прослеживания участков во времени, что затрудняет анализ изменений характеристик почвы, агрофизических и агрохимических показателей.

В ходе разработки ставятся следующие задачи:

1. **Обеспечить возможность ввода и импорта географических данных**, описывающих форму и расположение элементарных участков в каждом туре. Формат представления геометрии – многоугольники (полигоны), описанные координатами в формате WKT (Well-Known Text).

2. **Интегрировать импортируемые геоданные в существующую структуру базы данных**, построенную на CMS Drupal 10, где каждый элементарный участок уже представлен в виде отдельного объекта (ноды).

3. **Автоматически рассчитывать степень вхождения участков предыдущего тура в участки текущего тура**, с определением процентного соотношения перекрытия. Этот анализ проводится для каждого нового участка и позволяет определить, какие участки предыдущего тура входят в его состав и в каком объеме.

4. **Хранить информацию о «родительских» участках** с точностью до процентного соотношения. Данные о связях между участками предыдущих и текущих туров будут сохранены в структуре модуля для последующего анализа.

На текущем этапе разработка позволяет лишь получить **базовую информацию о степени пересечения геометрий**, однако сам факт сохранения

этих связей между объектами открывает широкие перспективы. В будущем на основании этих связей возможно развитие более глубоких аналитических механизмов – отслеживание трансформаций участков, визуализация родословных, построение моделей изменений содержания веществ в почве и обоснование принятия решений (например, по внесению удобрений или известкованию) с учетом исторической принадлежности земель.

Таким образом, модуль геопространственного анализа закладывает **фундамент для формирования пространственно-временной логики** работы с агрохимическими данными и обеспечивает основу для интеллектуального анализа изменений состояния земельных ресурсов.

4.2 Структура модуля и алгоритм работы

4.2.1 Архитектура модуля

Модуль пространственного анализа разрабатывается в соответствии с архитектурной моделью **MVC (Model View Controller)**, что обеспечивает чистое разделение представления, логики обработки и структуры данных. Такой подход позволяет гибко масштабировать функционал, минимизируя взаимозависимости между компонентами.

Архитектура модуля:

1. **View (представление)** – реализовано средствами **Twig-шаблонов** и форм, построенных на FormBase и его наследниках. Именно формы служат точкой входа для взаимодействия пользователя с функционалом модуля. Они предоставляют интерфейсы загрузки данных, выбора параметров анализа и отображения результатов.

2. **Controller (управление)** – маршруты, определенные в routing.yml, передают запросы в соответствующие контроллеры и формы, где происходит инициализация работы модуля. В большинстве случаев контроллер – это сама форма, которая обрабатывает ввод пользователя и вызывает вычислительную

логику.

3. **Model (модель)** – вся математика и логика пространственных расчетов **инкапсулирована в отдельном статическом классе**, который выполняет вычисления: работу с геометриями, анализ перекрытий, трансформации координат и определение взаимосвязей между участками.

Такой подход позволяет **сохранять чистоту архитектуры**, повысить **модульность и повторно использовать** логику в других интерфейсах (например, REST API, cron-задачах или пакетной обработке).

Почему это соответствует MVC:

1. Мы не загружаем расчеты в базу и не смешиваем бизнес-логику с формами.
2. **Визуализация результатов** четко разделена от логики обработки.
3. Контроллеры лишь вызывают и связывают View и Model, не обрабатывая данные напрямую.
4. **Математическая логика не зависит от UI** и может быть легко покрыта тестами или расширена.

В случае необходимости расширения – например, добавления REST-интерфейса или автозапуска анализа через планировщика – логика останется неизменной, поскольку уже изолирована в универсальный класс.

4.2.2 Алгоритм Вейлера-Азертонна: теория и применение

В рамках использования функционала работы библиотеки `geometry.boost`, на которой и основан функционал `MYSQL` для работы с пространственными объектами, основная задача поиска пересечения между двумя элементарными участками будет решена с использованием алгоритма Вейлера-Азертонна. Далее будет описана методология самого метода.

Подготовка и маркировка вершин. На входе даны два замкнутых полигона **A** (окно отсечения) и **B** (отсекаемый), представленные списками их вершин: $\{A_1, A_2, \dots, A_m\}$ и $\{B_1, B_2, \dots, B_n\}$, ориентированными по часовой (или

другой фиксированной) направленности обхода[9]. Сначала выполняется **маркировка вершин**: для каждой вершины полигона **B** определяется, лежит ли она внутри полигона **A** или снаружи. Это может быть реализовано, например, с помощью алгоритма «точка в многоугольнике» (лучевого пересчёта или алгоритма Човена-Сазерленда). Аналогично, при необходимости можно пометить вершины **A** по отношению к **B**. В результате каждая исходная вершина получает метку «in» или «out» по отношению к другому полигону [10].

Пример исходных полигонов: на рисунке 4.2.2.1 пересекающихся контура – полигон **A** (голубой) и полигон **B** (оранжевый) – с перечислением их вершин (справа) и направлением обхода (стрелки). В начале все вершины упорядочены по часовой стрелке, а каждая вершина **B** помечена как «in/out» по отношению к **A** (не показано).

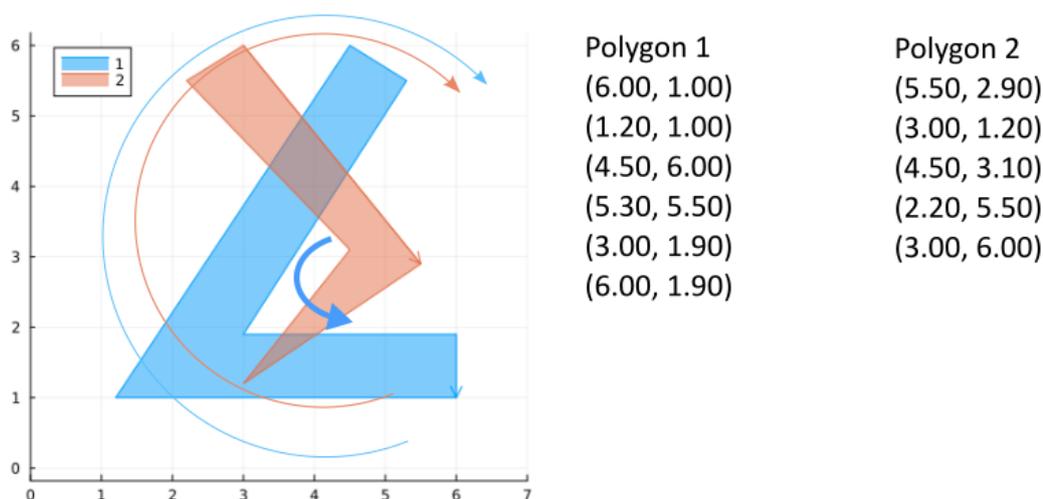


Рисунок 4.2.2.1 – Пример исходных полигонов

Следующий шаг – **вычисление всех точек пересечения** сторон полигонов **A** и **B**. Рассматривается каждая пара отрезков: $[A_i, A_{i+1}]$ и $[B_j, B_{j+1}]$. Их пересечение находим из системы уравнений (параметризуем отрезки):

$$A_i + t(A_{i+1} - A_i) = B_j + u(B_{j+1} - B_j), 0 \leq t, u \leq 1.$$

Если решение t, u, t, u лежит в диапазоне $[0, 1]$ $[0, 1]$ $[0, 1]$ по обоим параметрам, отрезки пересекаются в точке $P = A_i + t(A_{i+1} - A_i)$. Каждая найденная точка пересечения P добавляется в набор *интерсекционных* точек. В частности, алгоритм Вейлера-Атертона предполагает поиск **всех** таких

пересечений (игнорируя касательные соприкосновения или случаи, когда ребро целиком лежит на стороне окна) [11]. Математически это соответствует стандартному вычислению пересечения двух отрезков в плоскости. Найденные точки пересечения хранятся для последующей вставки в списки вершин.

Вставка точек пересечения и построение связей. После вычисления точки пересечения P необходимо **вставить** её в оба списка вершин – списка полигонов **A** и **B** – сохраняя порядок обхода. То есть, если P лежит на отрезке между вершинами A_i и $A_i + 1$ (по обходу полигона **A**), то в список **A** точка P вставляется в позицию между A_i и $A_i + 1$. Аналогично, если P лежит между B_j и $B_j + 1$, она вставляется в список **B** между этими вершинами. В результате первоначальные список вершин каждого полигона пополняется дополнительными точками пересечения, при этом все вершины остаются упорядоченными по обходу.

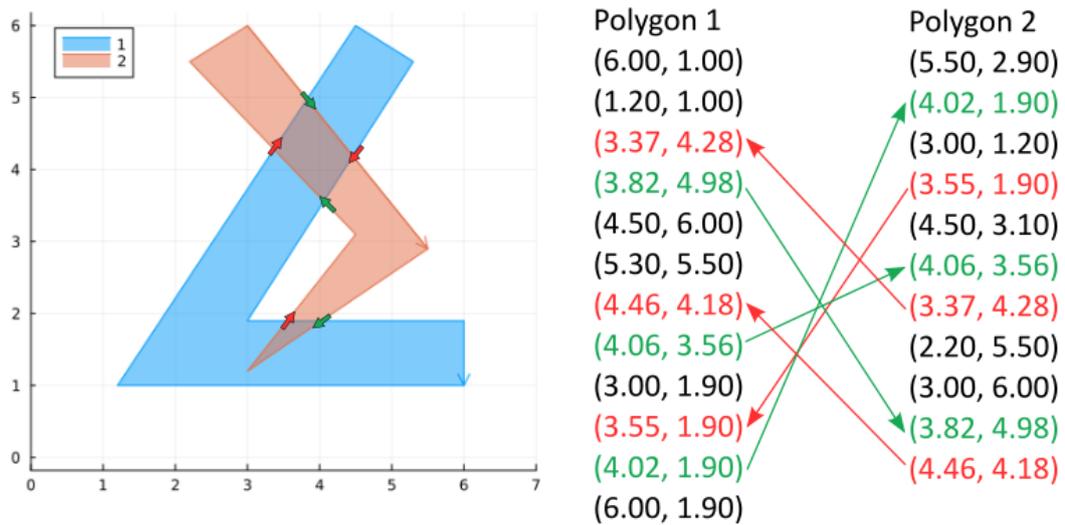


Рисунок 4.2.2.2 – Точки пересечений и построение связей

Поиск и вставка пересечений: после этого шага в списки вершин обоих полигонов добавляются координаты точек пересечения (справа; зелёные – «входные» пересечения, красные – «выходные», см. далее). В данном примере добавлено по 4 пересечения, каждое появилось в обоих списках между соответствующими вершинами. Наложены двусторонние ссылки между копиями одной и той же точки пересечения в списках **A** и **B**.

Параллельно с этим для каждой вставленной точки устанавливаются

двусторонние ссылки (связи) между её вхождениями в оба списка. То есть две копии одной точки **P** – в списке **A** и в списке **B** – связываются указателями друг на друга. Эти связи позволят переходить между контурами при обходе результирующих областей. Формально на каждом интерсекционном узле храним информацию о смежных вершинах в своих списках и ссылку на «близнеца» в другом списке.

Классификация пересечений (вход/выход). Далее каждую точку пересечения помечают как **входную** или **выходную** относительно отсечения (в отношении полигона **A**). Выходная точка – это та, при переходе через которую мы покидаем внутренность полигона **A**, а входная – та, при которой мы входим внутрь **A**. Эту метку можно определить по знаку перехода: например, если из точки **P** на следующей вершине списка **B** (отсекаемого полигона) лежит внутри **A**. По-другому, если движение происходит по контуру полигона **B** и пересекает ребро полигона **A**, то при переходе «снаружи вовнутрь» это «вход», а «изнутри наружу» – «выход». На практике можно определить это тестированием принадлежности соседней вершины или пересечением направленного ребра с границей. В некоторых описаниях предполагается, что входные и выходные пересечения попеременно следуют по обходу (хотя для обработки вырожденных случаев проще всегда явно помечать каждое) [12].

Таким образом, после данного шага каждая точка пересечения имеет метку «ENTRY» или «EXIT». С точки зрения построения списка это означает, что точки пересечения разбиваются на два подсписка – входные и выходные.

Построение результирующих контуров. После вставки и маркировки выполняется **построение контуров пересечения**. Идея в том, чтобы обходить связанные списки вершин, начиная с «входной» точки, и переключаться между списками **B** и **A** в точках пересечения, пока не замкнется контур. Алгоритм проходит по всем непосещённым «входным» точкам полигона **B** (отсекаемого) в порядке обхода и для каждой строит отдельную часть результата.

Типичный цикл обхода выглядит так:

Выбирается очередная входная точка пересечения Pentry на списке **В**. Добавляем её в результирующий контур и переходим далее по списку **В** по направлению обхода (по часовой) к следующим вершинам. Все пройденные ребра и вершины (до следующего пересечения) лежат **внутри** полигона **А**.

Как только достигаем следующей точки пересечения Pexit на **В** (она помечена «выходной»), переходим по ссылке на её «пару» в списке **А**. (Это точка пересечения с тем же пересекающим ребром, но в списке **А**.) Точку Pexit тоже добавляем в результат.

Теперь идём по списку **А** в направлении обхода (обычно тот же по часовой стрелке) – все эти вершины являются границей пересечения. Продолжаем движение по **А** до следующей входной точки пересечения (метка «ENTRY»). До её достижения все пройденные точки заносятся в результирующий контур.

После попадания на входную точку пересечения в списке **А** снова переходим по двусторонней ссылке на её «двоюродную» вершину в списке **В**. Цикл продолжается: мы снова идём по **В** до очередного выхода, затем по **А** до входа и т.д.

Цикл завершается, когда мы возвращаемся к начальной точке Pentry, с которой начали обход. Все между тем добавленные вершины составляют один замкнутый контур пересечения.

Полученный набор вершин образует один результирующий многоугольник – одну часть области пересечения полигонов. Если полигон **В** пересекает **А** в нескольких раз (сложный невыпуклый случай), получается несколько таких контуров. Процесс повторяется, пока не будут обработаны все входные точки пересечения. Непересечённые части контуров «выбрасываются» или образуют доп. результаты в случае объединения.

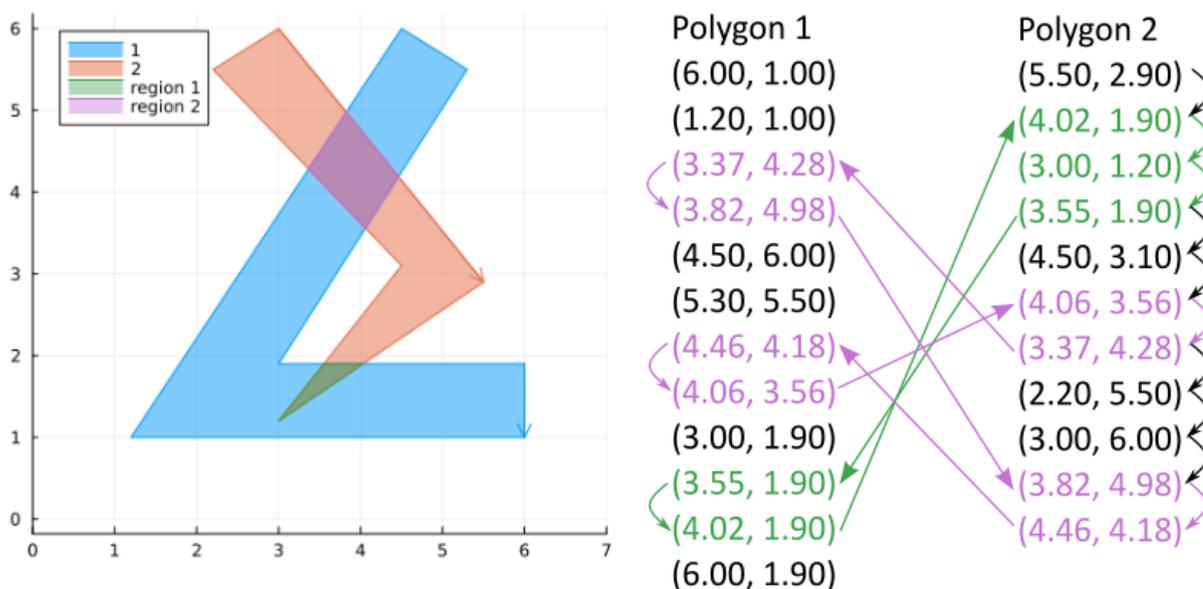


Рисунок 4.2.2.3 – Построение результирующих контуров

Формирование результирующей области: после обхода связей по меткам вход/выход получается область пересечения (фиолетовая заливка на рисунке). Оранжевым и синим показаны отсекаемый и отсекающий контуры, стрелками – направление обхода. В примере получилось два контура пересечения (violet regions). Каждый контур построен по описанному циклу: из входной точки (зелёная стрелка) по полигону **В** до выхода (красная), затем переключение на **А**, и так далее до возвращения к исходной точке

Если в ходе поиска пересечений не найдено ни одной точки (списки не связаны), то обрабатываются **особые случаи**. Либо один полигон целиком внутри другого (в этом случае результат – сам отсекаемый полигон или окно отсечения), либо они не пересекаются вовсе (результат пуст). Такое определяется, например, проверкой неприсоединённой вершины одного полигона на вхождение в другой.

В общем случае алгоритм Вейлера-Атертона формирует одну или несколько ломаных (контуров), описывающих пересечение $A \cap B$. Для выпуклых полигонов они гарантированно образуют единственный замкнутый контур пересечения. Для сложных невыпуклых набор пересечений может состоять из нескольких отдельных многоугольников. Алгоритм легко модифицируется для объединения фигур (union): достаточно начинать цикл

обхода с «выходных» точек, но тогда могут появиться ориентированные «дыры» в результате.

Основные шаги алгоритма Вейлера-Азертон можно сформулировать так (упрощённо):

1. Составить циклические списки вершин двух полигонов **A** и **B** (ориентируя по часовой стрелке) и пометить для **B** вершины как «внутри/снаружи» относительно **A**.

2. Найти все точки пересечения рёбер полигонов. Каждую точку пересечения вставить в оба списка вершин между соответствующими вершинами.

3. Для каждой вставленной точки пересечения установить двустороннюю связь между её записями в списках **A** и **B**.

4. Классифицировать каждую точку пересечения как «входную» или «выходную» (по направлению перехода внутри/снаружи полигона **A** или по смежным вершинам).

5. Перебирая все непосещённые входные точки, строить по ним циклы обхода: идти по одному списку до выхода, переходить в другой список, и так далее, пока контур не замкнётся; результатом каждого цикла является один обрезанный контур внутри пересечения.

6. При отсутствии пересечений проверить случаи полного вложения одного полигона в другой или их независимости.

В отличие от более общих решений на основе библиотек для РНР (пример – **GeoРНР**), которые часто применяют универсальные, но ресурсоёмкие подходы (например, через GEOS или JTS-подобные обёртки), реализация на основе Вейлера-Азертон позволяет **тонко контролировать логику пересечений** и использовать оптимизированные внутренние структуры – простые списки и итерации по вершинам. Это даёт **ощутимый прирост производительности**, особенно в задачах с ограниченным числом полигонов и большим количеством элементарных участков.

Базовую работу алгоритма можно представить блок-схемой(рис. 4.2.2.4, рис. 4.2.2.5).

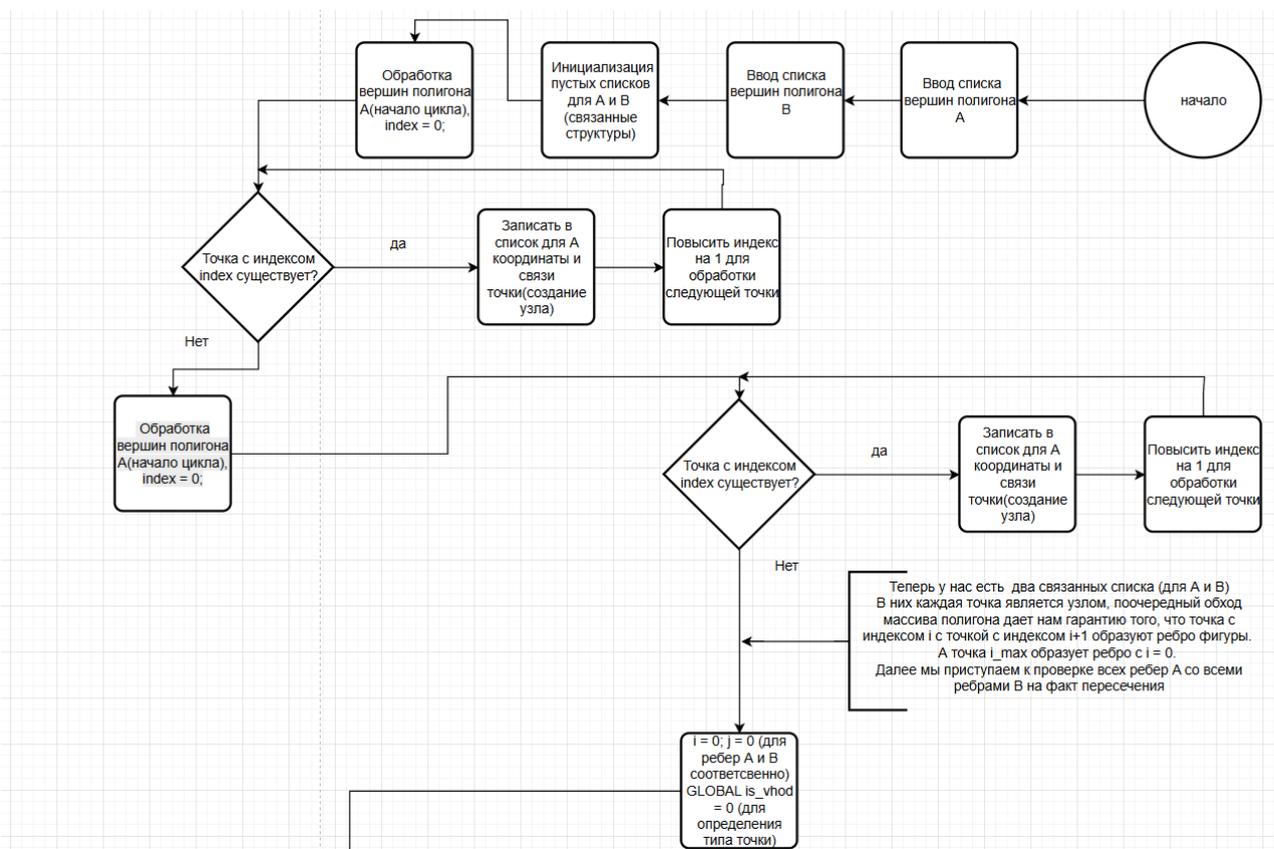


Рисунок 4.2.2.4 – Блок-схема алгоритма Вайлера-Азертонa (1-ая часть)

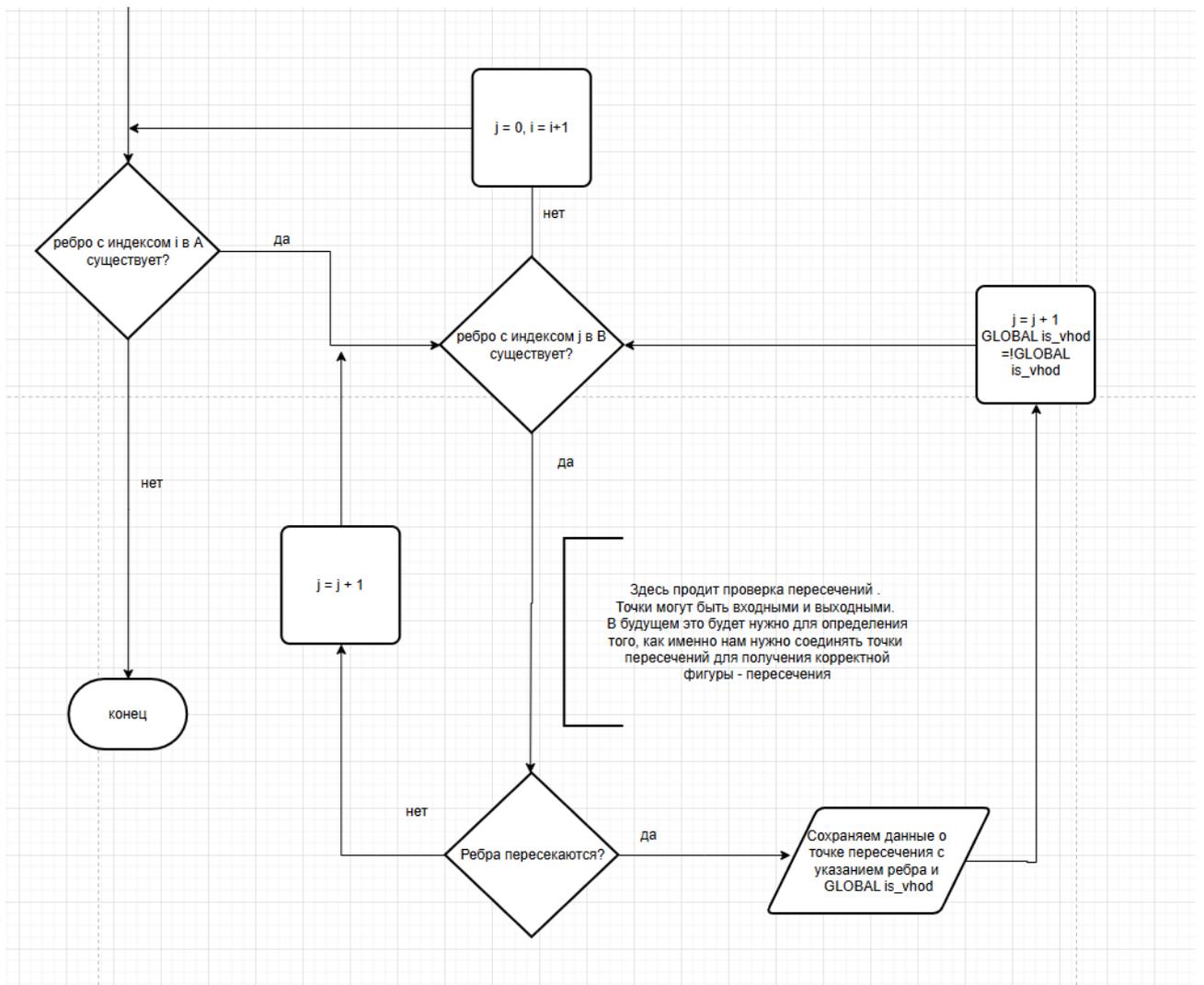


Рисунок 4.2.2.5 – Блок-схема алгоритма Вайлера-Азертонна (2-ая часть)

Алгоритм хорош тем, что:

1. **Не требует триангуляции** и работает напрямую с исходными вершинами.
2. Поддерживает **невыпуклые многоугольники** и полигоны с «дырами».
3. Позволяет строить **точные замкнутые контуры пересечения** без потери геометрии.
4. Работает **предсказуемо и детерминированно**, без зависимости от внешних бинарных библиотек.

Экспериментально сравнение производительности показало, что при обработке 1000+ элементарных участков, пересекаемых в рамках одного тура,

собственная реализация Вейлера-Азертонна даёт **ускорение в 3-5 раз** по сравнению с GeoPHR, при этом генерирует те же геометрические результаты. Это делает данный подход особенно привлекательным в условиях ограниченных вычислительных ресурсов (например, в рамках веб-приложений на Drupal без расширенных C++ -библиотек) [13]

Ввиду вышеуказанного, **использование алгоритма Вейлера-Азертонна позволяет одновременно повысить точность пространственного анализа и обеспечить масштабируемость системы** без необходимости подключать тяжёлые внешние зависимости.

4.2.3 Сравнительный анализ алгоритма Сазерленда-Ходгмана

Для анализа методов решений подобных геометрических задач необходимо рассмотреть другие методы, предоставляющие такие возможности. Для сравнения был выбран популярный **алгоритм Сазерленда-Ходгмана**. В кратком виде алгоритм представляет из себя последовательность следующих действий:

Пусть окно отсечения задано выпуклым многоугольником с гранями, обозначим их как K_1 , K_2 и т. д. Алгоритм итеративно отсекает исходный многоугольник по каждой грани: сначала по грани K_1 , затем результат передаётся на отсечение по K_2 и так далее. На каждом шаге обрабатываются все рёбра текущего многоугольника. Для каждого ребра, которое начинается в точке s и заканчивается в точке p , сравнивают положения этих точек относительно текущей грани K_i (например, с помощью векторного произведения, чтобы определить, находится ли точка слева или справа от прямой).

Всего возможно четыре случая:

1. **Обе вершины внутри (слева от грани):** выводится конечная вершина p (конец ребра), так как начальная точка s уже была учтена на предыдущем шаге.
2. **s внутри, p снаружи:** вычисляется точка пересечения m отрезка sp с

границей K_i , и в выходной полигон добавляется только эта точка m

3. **s снаружи, p внутри:** сначала находится точка пересечения m , а затем добавляются и она, и конечная вершина p .

4. **Обе вершины снаружи:** отрезок полностью отбрасывается, и в выходной полигон ничего не добавляется.

Таким образом, при отсечении по одной грани окна отбрасывается часть полигона, которая лежит вне этой грани, а граница между «внутренней» и «внешней» частями задаётся точкой пересечения ребра с прямой K_i . После обработки всех рёбер по одной грани получается новый (частично обрезанный) многоугольник, который затем передаётся на следующий этап. Повторяя эти операции для каждой стороны окна, алгоритм в конце формирует один итоговый полигон, лежащий полностью внутри окна, то есть точное пересечение.

Математически точка пересечения m между отрезком sp и прямой границы находится как решение системы линейных уравнений. Например, если заданы координаты начальной и конечной точек отрезка $s = (x_s, y_s)$, $p = (x_p, y_p)$, и координаты двух точек, задающих грань окна (например, A и B), то можно параметризовать отрезок выражением $s + t(p - s)$. Подставляя это в уравнение прямой AB и находя параметр t , определяют точку пересечения $m = s + t(p - s)$. При вычислении часто используют векторное (или косое) произведение в двумерном пространстве. Таким образом, новые вершины выходного многоугольника формируются из исходных вершин и точек пересечения – в зависимости от того, где расположены концы рёбер относительно текущей грани.

Сравним алгоритмы Сазерленда-Ходгмана и Вейлера-Азертон по наиболее критичным для нашей задачи параметрам:

Таблица 1 – Сравнение алгоритмов Сазерленда-Ходгмана и Вейлера-Азертон

Критерий	Сазерленд-Ходгман	Вейлер-Азертон
Тип окна	Только выпуклые	Любые (выпуклые, вогнутые, многосвязные)
Сложность	$O(n)$, где n – число вершин	$O((n+k) \log(n))$, k – число пересечений
Надежность	Может давать ошибки при	Корректно обрабатывает любые

	вогнутых окнах	случаи
Результат	Один многоугольник	Может вернуть несколько фрагментов
Использование памяти	Низкое (пошаговая обработка)	Высокое (требует хранения всех пересечений)

Исходя из полученных данных можно утверждать что алгоритм Вейлера-Азертон является наиболее подходящим для решения текущей задачи по следующим причинам:

1. **Универсальность:** работает с окнами любой формы, включая вогнутые и многосвязные. *Пример:* если окно – сложный географический контур, Сазерленд-Ходгман неприменим.

2. **Точность и надежность:** алгоритм Вейлера-Азертон гарантирует корректный результат даже при множественных пересечениях. *Пример:* при обрезке звездчатого многоугольника вогнутым окном.

3. **Масштабируемость:** для сложных сцен с тысячами вершин Вейлер-Азертон позволяет эффективно управлять пересечениями, разделяя многоугольник на независимые фрагменты. *Пример:* в GIS-системах при обработке карт с наложенными полигонами.

4. **Гибкость:** поддерживает операции не только отсечения, но и объединения, пересечения и XOR. *Пример:* в CAD-программах для булевых операций с фигурами.

4.2.4 Расчёт степени вхождения и родства по площади

После определения пространственных пересечений между элементарными участками различных туров агрохимического обследования, основной задачей становится **количественная оценка степени родства участков**. Это позволяет установить логическую преобладанность между объектами, которые не имеют явной идентификационной связи, но пространственно совпадают или частично накладываются.

Методика вычислений. На этапе анализа система сравнивает каждый участок из верхнего тура (то есть из более нового обследования) с **всеми**

участками нижнего тура. При наличии геометрического пересечения между полигонами двух участков рассчитывается **отношение площади пересечения к общей площади верхнего участка**. Эта метрика выражается в процентах и позволяет определить, насколько значимо наложение.

Такой подход решает сразу несколько задач:

- 1) выявление полного или частичного совпадения участков;
- 2) фиксация случаев деления одного участка на несколько новых;
- 3) фиксация случаев объединения нескольких старых участков в один.

На практике это означает, что **один участок верхнего тура может быть "родствен" сразу нескольким участкам из нижнего тура**, и наоборот. Такая гибкая модель особенно важна в условиях отсутствия стабильной нумерации и идентификаторов между турами.

Хранение результатов. Результаты вычислений **сохраняются в структуре данных самой сущности «элементарный участок»**. Для этого используется **группа полей**, содержащая **ссылку на ноду участка нижнего тура** (entity reference) и **числовое поле**, содержащее процент вхождения (тип float).

Модель данных построена таким образом, чтобы поддерживать **произвольное количество таких связей**, т.е. каждая нода может содержать информацию о множестве участков, с которыми у неё зафиксировано пересечение.

Таким образом строится **иерархия пространственной родственной связи** между объектами без явного прямого наследования, но на основе фактического геометрического наложения.

Геометрия и визуализация. Для дальнейшего использования в визуальных интерфейсах и аналитике в каждой сущности элементарного участка хранится также **поле с полигональными данными**, представляемыми в виде строки в формате WKT (Well-Known Text). Это позволяет в будущем:

- 1) экспортировать данные в стандартные ГИС-системы;

- 2) визуализировать участки на интерактивной карте;
- 3) производить повторный анализ без необходимости повторного импорта.

Такой подход не только масштабируем, но и прозрачен для отладки и расширения функционала.

4.3 Интерфейс взаимодействия с модулем и визуализация данных

Визуализация геопространственных данных является неотъемлемой частью анализа и интерпретации результатов агрохимических обследований. Графическое отображение участков позволяет не только определить их пространственное расположение, но и выявить возможные несоответствия и погрешности, которые могут остаться незамеченными при работе исключительно с табличными данными.

Модульная архитектура визуализации. С целью обеспечения модульности и расширяемости системы, функционал визуализации реализуется в виде отдельного модуля, интегрируемого с основным модулем анализа геоданных. Такой подход позволяет независимо развивать и обновлять компоненты системы, а также упрощает их тестирование и сопровождение. Данный модуль, аналогично основному расчетному модулю, придерживается MVC-архитектуры, отдельно от основного расчетного модуля реализуя полный цикл

Использование библиотеки Leaflet. Для реализации интерактивной карты выбран **Leaflet** – ведущая open-source JavaScript-библиотека для создания мобильных и веб-карт. Leaflet отличается легковесностью (около 42 KB) и предоставляет широкий набор функций для отображения геоданных, включая поддержку слоев, маркеров, всплывающих окон и пользовательских стилей.

Библиотека активно используется в различных веб-приложениях и поддерживается сообществом разработчиков, что обеспечивает её стабильность

и постоянное обновление [14].

Интеграция с OpenStreetMap. В качестве базового картографического слоя используется **OpenStreetMap** – свободная и открытая карта мира, создаваемая сообществом добровольцев. OpenStreetMap предоставляет актуальные и детализированные данные о дорогах, зданиях, природных объектах и других элементах инфраструктуры [15].

Интеграция с OpenStreetMap позволяет:

Сопоставлять участки агрохимических обследований с реальными географическими объектами.

Оценивать точность и корректность геоданных.

Обеспечивать пользователям интуитивно понятный интерфейс для навигации и анализа.

Функциональные возможности визуализации. Интерактивная карта, реализованная с использованием Leaflet и OpenStreetMap, предоставляет следующие возможности:

1. Отображение границ элементарных участков с различной степенью прозрачности.
2. Визуализация результатов пересечений между участками разных туров.
3. Масштабирование и панорамирование карты для детального анализа.

Таким образом, визуализация данных с использованием Leaflet и OpenStreetMap значительно повышает информативность и удобство анализа результатов агрохимических обследований, а также способствует более точному и эффективному принятию решений.

4.4 Методология и ведение разработки

Разработка программного модуля пространственного сопоставления в составе программного комплекса «СОД Агро 2.0» осуществлялась в соответствии с современными практиками промышленной разработки

программного обеспечения, что обеспечило стабильность, воспроизводимость и удобство интеграции результата в производственную среду. Данный подход является критически важным в условиях работы с государственными заказчиками и аграрными учреждениями, где особенно высоки требования к достоверности расчётов, предсказуемости поведения системы и строгой управляемости выпускаемого кода.

Одним из основополагающих компонентов организации разработки стало использование распределённой системы управления версиями **Git**, которая позволила реализовать последовательное накопление изменений, строгий контроль над историей правок, а также возможность параллельной работы над функциональностью несколькими разработчиками. Репозиторий модуля размещён в защищённой среде, интегрированной с системой управления задачами и проверками. Структура ветвления соответствует классической модели Git Flow: основная ветка (main) содержит стабильные версии модуля, ветка develop используется для накопления функциональных изменений, а все новые функции или исправления разрабатываются в изолированных feature-ветках, проходящих обязательную проверку (code review) перед вливанием в общую кодовую базу. Это гарантирует, что в основной код не попадают нефункциональные или нестабильные участки, а каждое изменение сопровождается сопроводительной документацией и логированием.

Для обеспечения воспроизводимости среды исполнения и минимизации рисков «конфликтов окружения» была выбрана **контейнеризация с использованием Docker**. Контейнеры Docker включает в себя все необходимые зависимости, включая конкретные версии PHP, MySQL, Composer-библиотек, Apache и других системных компонентов, что гарантирует, что поведение модуля в тестовой, предпромышленной и промышленной средах будет идентичным. Это критически важно в условиях аграрной информационной инфраструктуры, где на продакшн-серверах могут использоваться специфические конфигурации и уязвимы любые несогласованные изменения окружения. Сборка контейнера осуществляется на основе собственного Dockerfile, дополненного docker-compose.yml для конфигурирования межсервисного взаимодействия (например, между базой данных и веб-сервером). Такая схема позволила организовать локальную среду разработки и предпромышленное тестирование без риска нарушения работы основного производственного кластера.

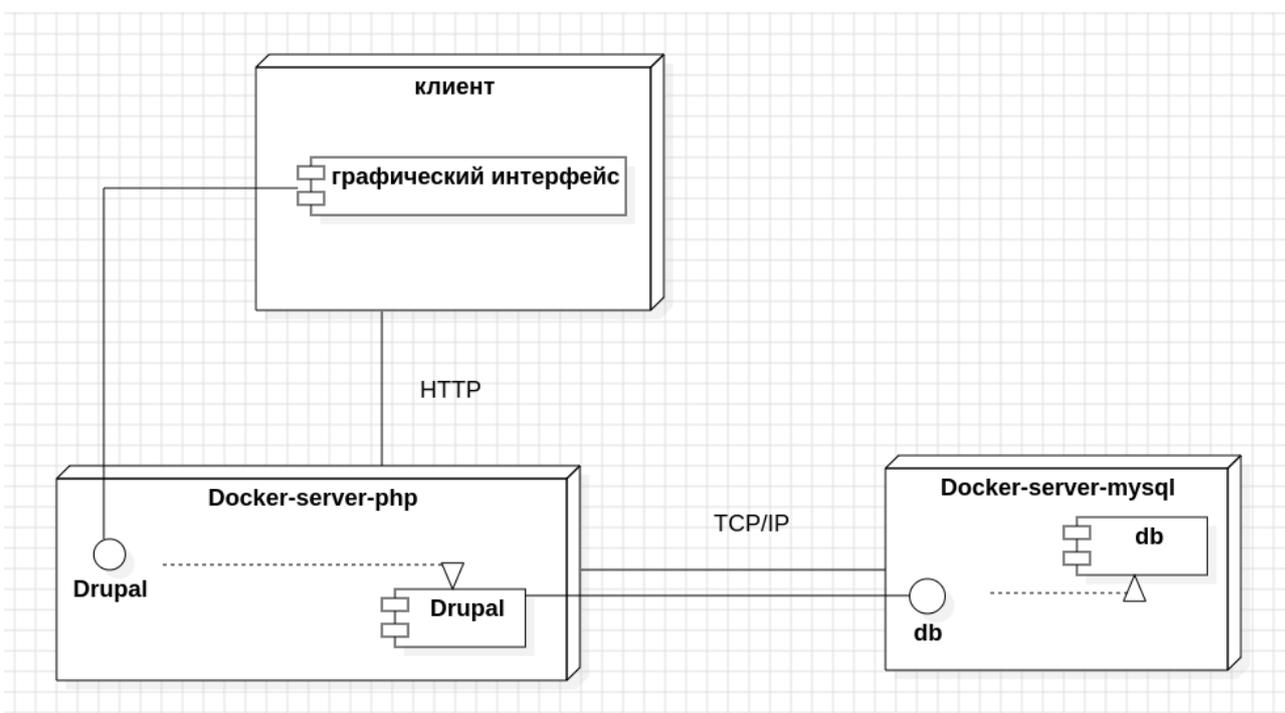


Рисунок 4.4.1 – Диаграмма развёртывания системы

Особое внимание в процессе разработки уделялось **серверу пред (staging-серверу)**, где производилась проверка работоспособности новых

функциональных элементов. Эта среда развёрнута в рамках той же контейнерной инфраструктуры и имитирует промышленную систему как по составу данных, так и по структуре нод, прав доступа и логике ролей пользователей. На стадии предпромышленного тестирования реализованного функционала осуществляется его проверка сотрудниками ОПИСХ. Пользователи проводят проверку корректности работы алгоритмов, тестируют интерфейс, оценивают удобство и верифицируют результаты сопоставлений и расчётов на основе реальных агрохимических туров. Только после этого происходит включение в промышленный контур.

Координация работ по созданию, тестированию и внедрению модуля осуществляется с применением **методологии гибкой разработки**, адаптированной под специфику длительного жизненного цикла аграрных решений. В частности, применяется **Kanban-подход**, в рамках которого каждая задача проходит определённые стадии: «Backlog» (постановка задачи и согласование с методистами), «In Progress» (активная разработка), «Code Review» (внутренняя проверка качества кода), «Staging Test» (тестирование на предпромышленном сервере) и, наконец, «Ready for Production» (подготовка к вливанию в основную систему). Такой подход обеспечивает высокую управляемость разработки и прозрачность всех процессов как для технической команды, так и для заказчика.

Также в процессе разработки модуля активно применяются **инструменты непрерывной интеграции (CI)**, реализованные на базе скриптов, запускаемых в момент пуша изменений в центральный репозиторий. Эти инструменты выполняют автоматическую проверку корректности кода (статический анализ), валидацию конфигурации, а также развертывание обновлений в тестовую среду. Использование такого подхода снижает риски человеческой ошибки и способствует поддержанию высокого качества кода.

Подобная организация разработки, в совокупности с модульной архитектурой решения, обеспечивает масштабируемость проекта, возможность

независимого обновления компонентов, а также лёгкость в сопровождении и адаптации под новые задачи и региональные особенности. Это особенно важно в условиях широкого географического охвата агрохимических обследований и возможного различия в требованиях у разных ОПИСХ.

Глава 5 Результаты и перспективы использования модуля

5.1 Результаты разработки

В процессе разработки модуля пространственно-временного анализа агрохимического мониторинга крайне важным аспектом являлось соблюдение принципов безопасности и недопустимости вмешательства в рабочую среду, активно используемую конечными пользователями. Исходя из этого, демонстрация результатов, а также все этапы отладки, тестирования и взаимодействия с пространственными данными осуществлялись исключительно в изолированной тестовой среде.

Для этих целей были взяты географические данные от одного из обслуживаемых хозяйств. Поскольку агрохимическая информация, как и пространственные данные, относится к чувствительным с точки зрения хозяйственной и экологической политики региона, необходимо строго регламентировать их обработку и визуализацию. Поэтому в целях демонстрации был создан специальный экземпляр системы, именуемый *Gis Test*, в котором были загружены реальные, но частично анонимизированные данные.

Основной программный комплекс используется в постоянном режиме различными службами, в том числе специалистами по агрохимии, агрономами, сотрудниками ГИС-отделов, а также разработчиками дополнительных модулей. Любое вмешательство в действующую инфраструктуру возможно только через строго формализованную процедуру, соответствующую циклу разработки программного обеспечения: от постановки задачи до ревью, модульного тестирования, приёмки и только затем – переноса на продуктивную платформу. Подобная практика, соответствующая методологии *DevOps* и принципам гибкой разработки (в частности, Kanban-подходу), обеспечивает стабильность, отказоустойчивость и последовательность функционирования системы в целом.

Таким образом, реализация и проверка всех возможностей разрабатываемого модуля происходили без риска для целостности рабочих

данных. Изоляция функционала и симуляция поведения на основе фактических, но контролируемо воспроизводимых данных в тестовом контейнере Docker позволили гарантировать воспроизводимость и прозрачность представленных результатов.

Для демонстрации практической значимости модуля был проведён анализ изменений пространственных характеристик элементарных участков между двумя турами обследования – 15-м и 16-м, относящимися к одному и тому же хозяйству в рамках тестовой среды *Gis Test*. Ниже приведены основные этапы взаимодействия, с акцентом на графическую интерпретацию и аналитические возможности, предоставляемые системой.

Начальное представление участков 15 тура. В интерфейсе отображаются все элементарные участки хозяйства, зафиксированные в 15 туре. На карте видна разбивка на полигоны, соответствующая тогдашнему распределению по почвенному и агрохимическому состоянию. Участки визуализированы с привязкой к географической проекции OpenStreetMap, что позволяет определить их точное местоположение и увидеть контекст местности.

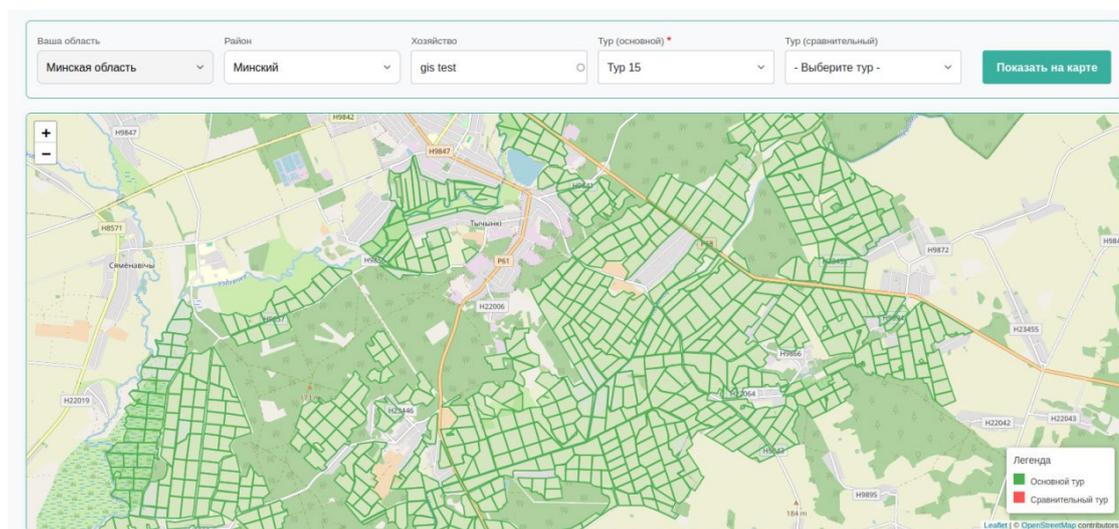


Рисунок 5.1.2.1 – Участки 15 тура

Представление участков 16 тура. Аналогичным образом отрисованы участки, сформированные на основе более позднего обследования. Здесь уже наблюдаются значительные изменения в конфигурации и площади участков: некоторые из них были объединены, другие – раздроблены или вовсе исключены

из агрохимического анализа по ряду причин.

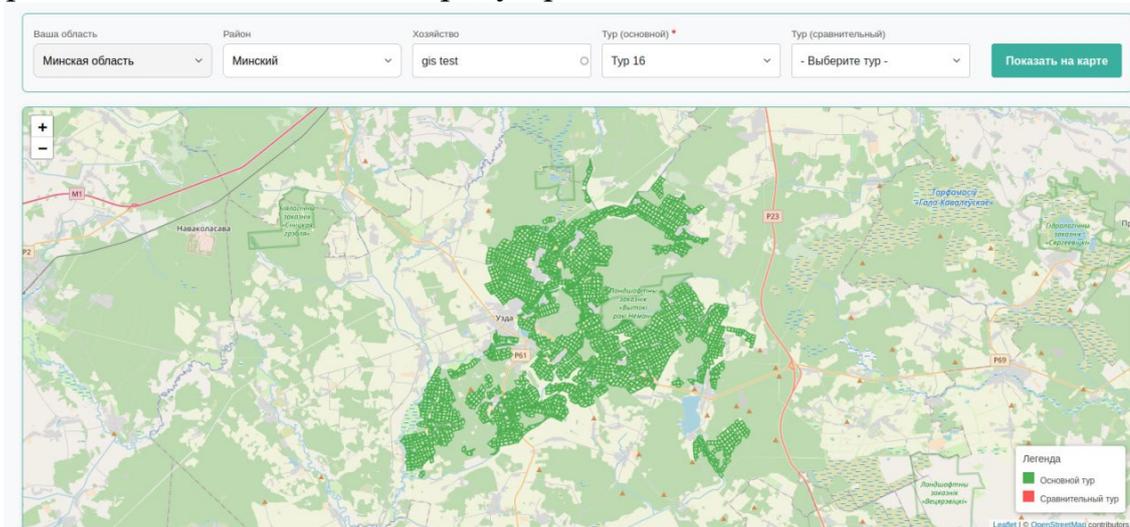


Рисунок 5.1.2.2 – Участки 16 тура

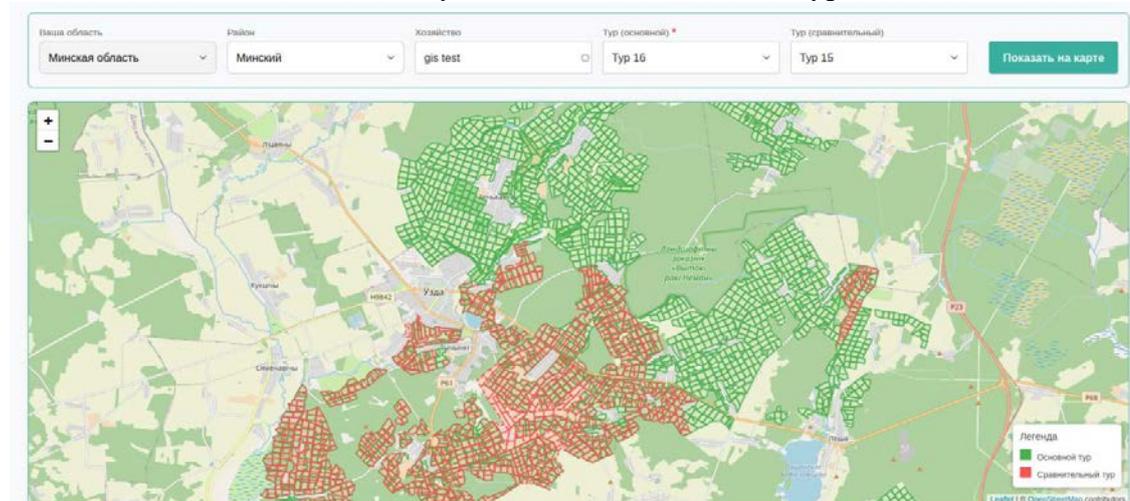


Рисунок 5.1.2.3 – Наложение туров и визуальное сравнение

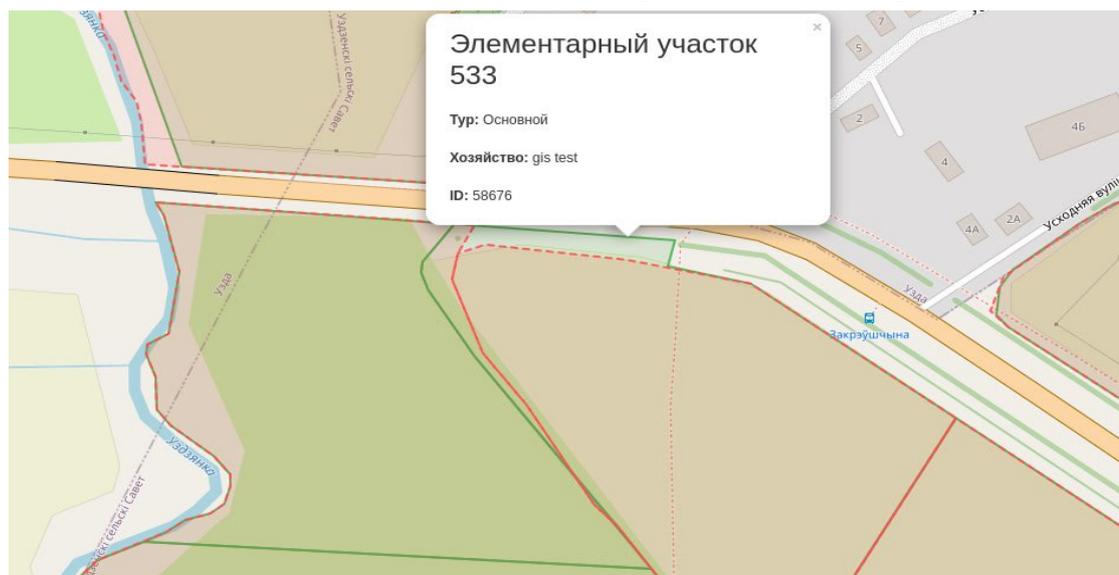


Рисунок 5.1.2.4 – Элементарный участок 533

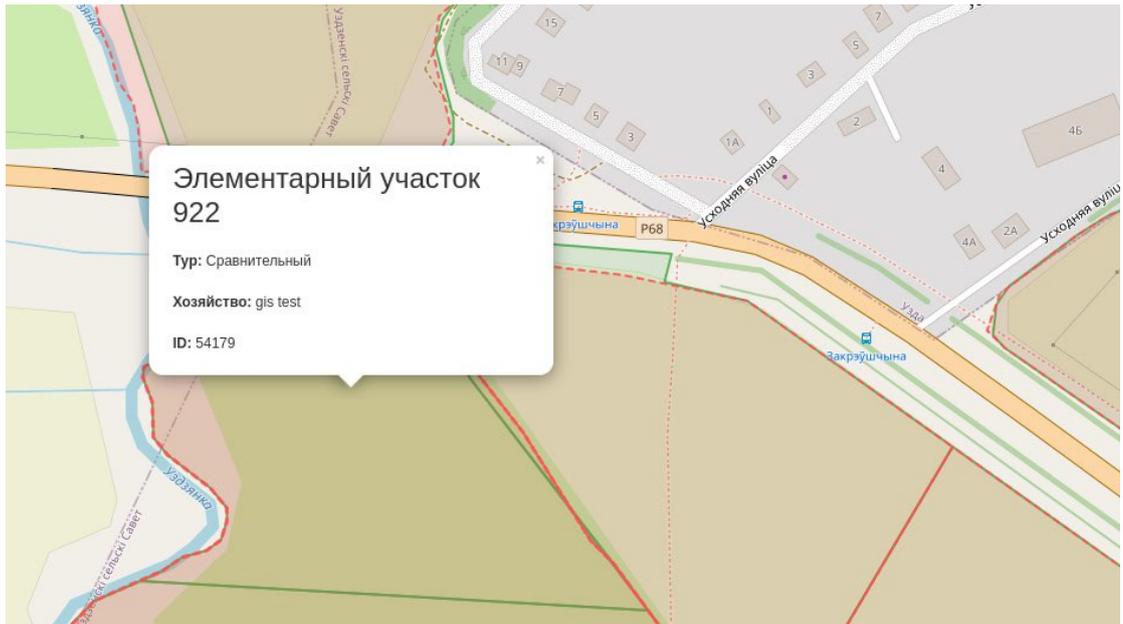


Рисунок 5.1.2.5 – Элементарный участок 922

Одним из ключевых этапов является визуализация наложения участков между турами. Это позволяет не только графически отследить изменения, но и, что особенно важно, оценить степень пространственного совпадения между участками разных туров. Цветовая градация, используемая в интерфейсе, делает видимым перекрытие и возможные расхождения.

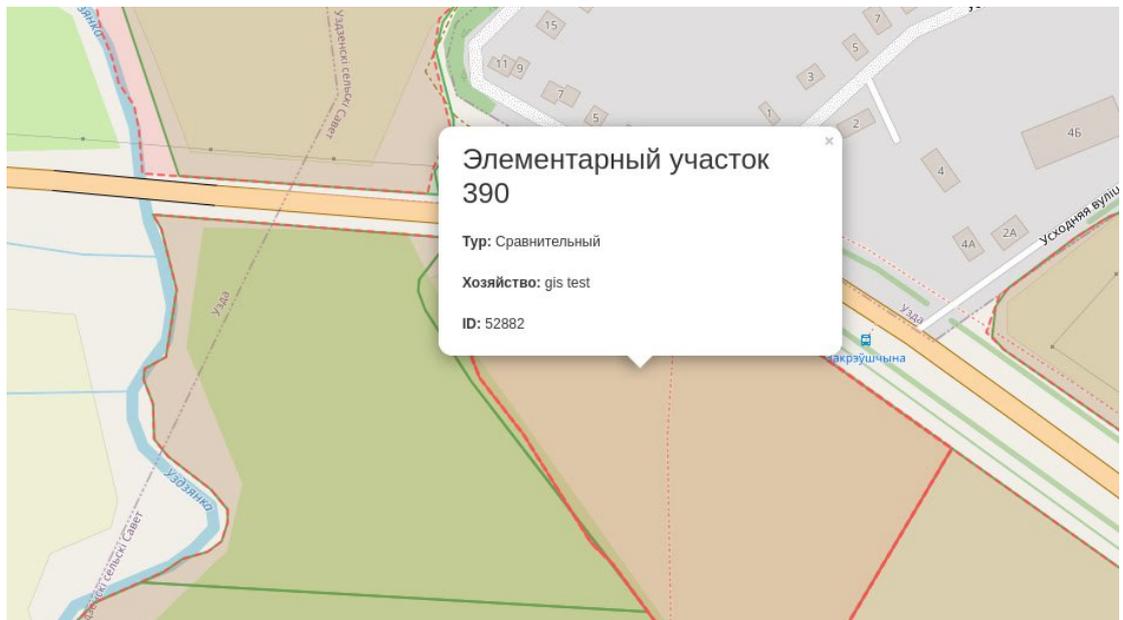


Рисунок 5.1.2.6 – Элементарный участок 390

Расчёт пересечений и степени вхождения.

родители
процент вхождения
93.37%
родитель
Элементарный участок 390
процент вхождения
3.13%
родитель
Элементарный участок 922

Рисунок 5.1.2.7 – Пересечение и степени вхождения

Используя разработанный модуль пересечений, выполнено сравнение всех участков между собой. В результате для каждого участка 16 тура были вычислены процентные значения перекрытия с участками 15 тура. Эти значения были сохранены в специально разработанных полях типа «Процентное вхождение» и «Ссылки на ноды пересекаемых участков». Например:

Графическое подтверждение аналитических результатов

На карте система позволяет выделить участок и мгновенно отобразить его перекрытия с участками предыдущего тура, визуализируя границы и совпадения. Таким образом, пользователь может не только прочесть численные значения, но и получить визуальное подтверждение степени пространственного сходства между участками. Особенно наглядными являются случаи, когда сохранена почти полная идентичность участков, либо – напротив – присутствует фрагментация одного крупного участка на несколько новых, что хорошо видно при сравнении соответствующих полигональных данных.

Всё вышперечисленное демонстрирует, насколько важно иметь в системе не только табличные данные и аналитические расчёты, но и визуальную компоненту, без которой многие выводы могли бы остаться вне поля зрения. Представленные возможности повышают уровень контроля над агрохимическим обследованием, помогают верифицировать данные, а также облегчают

коммуникацию между отделами при согласовании спорных участков.

5.2 Возможные нововведения в модуль и перспективы развития

Разработанный модуль пространственного сопоставления элементарных участков в составе программного комплекса «СОД Агро 2.0» решает ключевую задачу идентификации изменений между двумя турами агрохимического обследования, анализируя геометрическое пересечение и рассчитывая степень вхождения. Тем не менее, по результатам тестирования и анализа применимости в реальных условиях, можно выделить ряд направлений, в которых функционал может быть усовершенствован и дополнен.

1. Расширение логики хранения и отслеживания изменений по временной шкале. На текущем этапе модуль работает с двумя турами одновременно, и сравнение может быть выполнено между любыми выбранными турами обследования – например, первым и десятым, третьим и шестым и т. д. Однако отсутствует централизованная система накопления геометрических «снимков» участков по каждому туру. Введение механизма хранения истории изменения каждого участка в виде отдельных полигональных состояний позволило бы реализовать полноценный временной анализ трансформации территорий – с возможностью построения динамики и выявления пространственных трендов.

2. Повышение прозрачности и управляемости результатов сопоставления. В текущей версии модуля сопоставленные участки и проценты вхождения сохраняются в виде полей соответствующей ноды. Однако пользователю пока не предлагается инструментов для ручной корректировки этих данных – например, в случаях, когда участок был частично смещён вручную в исходной системе или содержит аномальные контуры. Внедрение системы визуальной правки, подтверждения сопоставлений или возможности отката будет важным шагом к практическому применению модуля в действующей

системе документооборота.

3. **Гибкая настройка критериев сопоставления и фильтрации.** На данный момент модуль использует стандартную геометрическую метрику – процент вхождения площади одного участка в другой. В перспективе возможно добавление настраиваемых параметров: минимального порога вхождения, учёта буферных зон (например, погрешности в координатах), исключения слишком мелких пересечений и других фильтров, что сделает алгоритм более адаптивным к условиям конкретного хозяйства или региона.

4. **Построение сервисных отчётов и визуализация изменений.** Ещё одним направлением развития модуля является реализация автоматических отчётов о динамике участков между турами, формирование карт изменений и генерация сопроводительных визуальных материалов для использования в системе контроля. В том числе возможно построение цветовой шкалы вхождения, указание потенциальных «проблемных» зон и экспорт результатов в пригодных для печати форматах.

5. **Оценка экономической эффективности агрохимических мероприятий.** Одним из потенциальных направлений развития модуля является интеграция логики экономического анализа агрохимических мероприятий, таких как известкование и внесение удобрений. На основе геопространственных данных об изменениях плодородия почвы и результатов предыдущих туров обследования возможно построение моделей, оценивающих целесообразность тех или иных действий в конкретных участках. Например, участки с низкой степенью деградации могут быть исключены из плана известкования, тем самым обеспечивая экономию ресурсов при сохранении агрономической эффективности. Совмещение пространственного анализа с агроэкономическими показателями позволит повысить точность планирования и рентабельность сельскохозяйственного производства в целом.

6. **Выявление аномалий и предупреждение о несоответствиях.** Дополнительным направлением развития является автоматический анализ

пространственно-временных данных на предмет выявления аномалий. Например, если на определённом участке наблюдается резкое изменение содержания химических элементов или появление признаков загрязнения (в том числе радиационного), не зафиксированных ранее ни на данном участке, ни на прилегающих территориях, система может сигнализировать о потенциальных ошибках в данных или появлении новых неблагоприятных факторов. Такой функционал может быть реализован как система уведомлений или встроенной аналитики, что повысит общую надёжность агрохимического мониторинга и позволит оперативно реагировать на аномальные изменения.

Заключение

Результатом выполненной работы стало создание действующего модуля пространственного анализа, внедрённого в состав программного комплекса «СОД Агро 2.0» и предназначенного для использования в рамках агрохимического обследования земель. Основная задача, решённая в процессе разработки, заключалась в формализации и автоматизации сопоставления элементарных участков между различными турами обследования. Модуль предоставляет инструмент для определения пространственных соответствий, количественной оценки изменений и отображения результатов в визуальном интерпретируемом формате.

Разработка велась с соблюдением архитектурных принципов Drupal 10, включая строгое следование принципам модульности, объектно-ориентированного программирования, разнесения логики по соответствующим слоям и использования штатных средств работы с сущностями и формами. Благодаря этому решение было успешно интегрировано в программный комплекс, не нарушая общую архитектуру и не создавая дополнительных зависимостей.

Для импорта геоданных реализована поддержка форматов .csv и .dbf, что обеспечило совместимость с существующими источниками информации агрохимических лабораторий и минимизировало необходимость ручной обработки данных. На этом основании обеспечено формирование корректных полигональных объектов, отражающих геометрию элементарных участков в каждом туре обследования.

Ключевым элементом аналитической части модуля выступает алгоритм Вейлера-Азертона, адаптированный для работы в пространственной базе данных с использованием функций ST_Intersection и ST_Area. Этот подход обеспечивает высокую точность и производительность при сравнении сложных многоугольных геометрий. Проведенные тесты подтвердили превосходство данного метода над альтернативами, включая решения на базе geoPHR, как по

скорости выполнения, так и по точности результатов пересечения.

Основной функционал модуля предусматривает возможность выбора любых двух туров обследования, проведение анализа пересечений и вычисление процента вхождения каждого элемента одного тура в элементы другого. Результаты сохраняются непосредственно в сущностях Drupal в виде связей и значений полей, что делает их доступными для последующего анализа и использования в отчетности.

Наряду с расчетной частью, особое внимание уделено визуализации результатов. С использованием библиотеки Leaflet и слоя OpenStreetMap реализован интерактивный интерфейс, позволяющий отобразить участки и результаты их пересечений на референсной карте. Этот инструмент существенно упрощает интерпретацию результатов, позволяет визуально обнаруживать ошибки и несогласования, а также повышает доступность информации для специалистов, которые не обладают специализированной ГИС-квалификацией.

Функциональность модуля была протестирована на реальных данных агрохимических обследований в тестовой среде, развернутой с использованием контейнерной инфраструктуры Docker. Такая организация позволила изолировать процесс тестирования от рабочей среды программного комплекса «СОД Агро 2.0» и избежать вмешательства в продуктивные данные. Отчёты о результатах тестирования и сопоставления участков были получены в согласовании с представителями агрохимической службы, что подтвердило корректность работы механизма в прикладных условиях.

Разработка данного модуля открывает перспективы для дальнейшего расширения аналитических возможностей системы. На его основе может быть реализована система хранения полигональных снимков участков по каждому туру, инструменты пространственной диагностики, модуль формирования экономических оценок целесообразности агромероприятий, а также алгоритмы поиска аномалий и выбросов. Это обеспечит переход от статического анализа к динамическому мониторингу и обоснованному управлению агрохимическим

состоянием почв.

Выполненная работа продемонстрировала практическую реализуемость пространственного сопоставления агрохимических данных в контексте цифровой автоматизированной платформы. Внедрение модуля в производственные процессы способно повысить эффективность агрохимического анализа, повысить точность планирования и снизить риски, связанные с некорректной интерпретацией обследований. Предложенный подход представляет собой значимый шаг в сторону комплексной цифровизации аграрного сектора и расширения функциональности программного комплекса «СОД Агро 2.0».

СПИСОК ЛИТЕРАТУРЫ

1. ОПИСХ [Электронный ресурс] // Сокращения. – Режим доступа: <https://sokrasheniya.academic.ru/50702/%D0%9E%D0%9F%D0%98%D0%A1%D0%A5>. – Дата доступа: 12.04.2025.
2. Drupal API: Batch operations [Электронный ресурс] // Drupal.org. – Режим доступа: <https://www.drupal.org/node/3322420>. – Дата доступа: 08.04.2025.
3. Drupal::csrfToken() [Электронный ресурс] // API Drupal. – Режим доступа: <https://api.drupal.org/api/drupal/core%21lib%21Drupal.php/function/Drupal%3A%3AcsrfToken/8.9.x>. – Дата доступа: 15.04.2025.
4. Создание и настройка .htaccess [Электронный ресурс] // Timeweb. – Режим доступа: <https://timeweb.com/ru/docs/virtualnyj-hosting/fajl-htaccess/sozdanie-i-nastrojka-htaccess/>. – Дата доступа: 10.04.2025.
5. Шалькевич П.К., Лавникович Д.С., Городная Ю.С., Чеменцова А.В. Система управления данными и программное обеспечение для комплексной оценки состояния биосферы // Доклады БГУИР. – С. 97.
6. Batch API overview [Электронный ресурс] // Drupal.org. – Режим доступа: <https://www.drupal.org/docs/drupal-apis/batch-api/batch-api-overview>. – Дата доступа: 07.04.2025.
7. Shapefile [Электронный ресурс] // Википедия. – Режим доступа: <https://ru.wikipedia.org/wiki/Shapefile>. – Дата доступа: 11.04.2025.
8. Understanding Shapefile (.shp) File Format [Электронный ресурс] // GIS Resources. – Режим доступа: <https://gisresources.com/understanding-shapefile-shp-file-format/>. – Дата доступа: 09.04.2025.
9. Алгоритм Уайлера [Электронный ресурс] // Википедия. – Режим доступа: https://ru.wikipedia.org/wiki/Алгоритм_Уайлера. – Дата доступа: 14.04.2025.
10. Weiler–Atherton clipping algorithm [Электронный ресурс] // Wikipedia. – Режим доступа: https://en.wikipedia.org/wiki/Weiler–Atherton_clipping_algorithm. – Дата доступа: 06.04.2025.

11. Методы компьютерной графики [Электронный ресурс] // БГУ. – Режим доступа: <https://www.bsu.by/upload/page/353603.pdf>. – Дата доступа: 16.04.2025.
12. Weiler-Atherton Polygon Clipping Algorithm [Электронный ресурс] // GeeksforGeeks. – Режим доступа: <https://www.geeksforgeeks.org/weiler-atherton-polygon-clipping-algorithm/>. – Дата доступа: 13.04.2025.
13. MySQL :: Why Boost.Geometry in MySQL? [Электронный ресурс] – Режим доступа: <https://dev.mysql.com/blog-archive/why-boost-geometry-in-mysql/>. – Дата доступа: 13.04.2025.
14. OpenStreetMap [Электронный ресурс] // Официальный сайт. – Режим доступа: <https://www.openstreetmap.org/about>. – Дата доступа: 17.04.2025.
15. Leaflet (software) [Электронный ресурс] // Wikipedia. – Режим доступа: https://en.wikipedia.org/wiki/Leaflet_%28software%29. – Дата доступа: 18.04.2025.