

Министерство образования и науки Российской Федерации

ТОМСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ  
СИСТЕМ УПРАВЛЕНИЯ И РАДИОЭЛЕКТРОНИКИ

**Ю.Б. Гриценко, Ю.П. Ехлаков, О.И. Жуковский**

**ГЕОИНФОРМАЦИОННЫЕ ТЕХНОЛОГИИ  
МОНИТОРИНГА ИНЖЕНЕРНЫХ СЕТЕЙ**

Томск  
Издательство ТУСУРа  
2010

УДК 625.78:91:004

ББК 38.788с51

Г858

**Рецензенты:**

**В.Ф. Тарасенко**, д-р техн. наук, профессор кафедры системного анализа и управления Томского государственного университета;

**С.Н. Колупаева**, д-р физ.-мат. наук, проректор по информатизации, зав. кафедрой прикладной математики Томского государственного архитектурно-строительного университета

**Гриценко Ю.Б.**

Г858 Геоинформационные технологии мониторинга инженерных сетей: монография / Ю.Б. Гриценко, Ю.П. Ехлаков, О.И. Жуковский. — Томск : Изд-во Томск. гос. ун-та систем управления и радиоэлектроники, 2010. — 148 с.

ISBN 978-5-86889-542-5

Рассматриваются вопросы реализации проектов по созданию АИС на основе геоинформационных технологий сопровождения инженерных сетей. Внимание акцентируется как на развитии методов моделирования, применяемых для представления инженерных сетей, так и на совершенствовании подходов к организации геоинформационных систем, ориентированных на информационное сопровождение электронных генеральных планов. Представлены архитектурные решения, использованные авторским коллективом при создании веб-ориентированных ГИС для информационной поддержки процесса ведения инженерных сетей предприятий металлургической и химической отраслей.

Для специалистов в области автоматизации управления инженерными сетями, разработки и создания ГИС, а также студентов вузов специальностей, связанных с проектированием и созданием современных информационных систем.

УДК 625.78:91:004

ББК 38.788с51

Научное издание

**Гриценко Юрий Борисович, Ехлаков Юрий Поликарпович,  
Жуковский Олег Игоревич**

**ГЕОИНФОРМАЦИОННЫЕ ТЕХНОЛОГИИ МОНИТОРИНГА ИНЖЕНЕРНЫХ СЕТЕЙ**

Монография

Корректор О.В. Полещук

Подписано в печать 25.12.10. Формат 60x84/16. Усл. печ. л. 8,6. Тираж 100 экз. Заказ 1205.

Томский государственный университет систем управления и радиоэлектроники  
634050, г. Томск, пр. Ленина, 40. Тел. (3822) 53-30-18.

ISBN 978-5-86889-542-5

© Гриценко Ю.Б., Ехлаков Ю.П.,  
Жуковский О.И., 2010

© Томск. гос. ун-т систем управления  
и радиоэлектроники, 2010

Инженерные сети, являясь основой инженерной инфраструктуры, в значительной мере определяют устойчивость развития как промышленных, так и социально-экономических территориальных образований. Надежность функционирования, качество транспортировки технологических и энергетических продуктов при эксплуатации инженерных сетей зависят от эффективности проведения процесса мониторинга их текущего состояния.

В предлагаемой работе авторами проведено обобщение практических результатов, полученных за последние десять лет в процессе работы по созданию автоматизированных информационных систем на основе геоинформационных технологий для сопровождения инженерных сетей таких объектов, как водоканалы городов Томск и Северск, Кузнецкий металлургический комбинат (ОАО «КМК», г. Новокузнецк), Томский нефтехимический комбинат (ОАО «Томскнефтехим»).

*В первой главе* дается общее представление об инженерных сетях. Рассматриваются некоторые подходы к их классификации. Особое внимание уделено жизненному циклу инженерной сети и характеристикам его этапов, формулированию задач информационной поддержки этапов, при этом акцент сделан на задачах этапа эксплуатации сети. Рассмотрены классификация задач, решаемых в процессе эксплуатации территориально распределенных инженерных сетей, и проблемы их автоматизации.

Несмотря на значительные различия в типах транспортируемых продуктов и технологиях, используемых различными эксплуатирующими организациями, современные инженерные сети характеризуются рядом общих специфических особенностей, что позволяет ставить и решать определенные прикладные задачи управления инженерными сетями, представленные в первой главе, при этом наиболее подробно рассматриваются задачи информационного обеспечения управления и анализа инженерных сетей.

*Во второй главе* рассматриваются проблемы моделирования инженерных сетей. Приводятся известные подходы построения плоских и многоуровневых моделей, предлагается формальное описание процессов взаимодействия объектов на различных уровнях инженерной сети, причем особо акцентируется внимание на вопросах унификации параметров

и функций предлагаемой алгебраической модели инженерной сети. Рассмотрены особенности моделирования жизненного цикла объектов инженерной сети, проанализированы подходы к представлению структур данных для решения задач моделирования процессов транспортировки продукта внутри сети.

*В третьей главе* анализируются особенности использования геоинформационных систем в процессе управления инженерными сетями с учетом общей организации таких систем, а также особенности использования настольных, сетевых и веб-ориентированных ГИС. Приведен анализ информационных технологий, используемых при построении веб-ориентированных ГИС. Рассматриваются общие принципы архитектурных решений, применяемых при создании ГИС. Кроме того, предложена технология хеширования данных для сопоставления графического и атрибутивного описаний объектов и рассмотрены пути решения проблемы разграничения прав доступа к данным в системах на основе пространственной базы данных.

*Четвертая глава* содержит анализ проблем, возникающих при создании веб-ориентированной ГИС инженерного назначения. Рассмотрена многоуровневая архитектура веб-ориентированных ГИС, отдельное внимание уделено выбору архитектуры создаваемых приложений как на стороне сервера, так и на стороне клиента. Предлагается модифицированный сервис-ориентированный подход для построения геоинформационной системы. Раскрываются особенности применения Mashup-идеологии, показана специфика сервис-ориентированной архитектуры блока анализа веб-ориентированной ГИС.

Опыт создания инженерных геоинформационных систем представлен *в пятой главе*. Приводится описание основных особенностей одной из последних версий веб-ориентированной геоинформационной системы, разработанной под руководством авторов данной монографии. Рассматриваются вопросы администрирования и управления данными в системе, приводится описание основных функций.

Среди сотрудников ТУСУРа, внесших наибольший, а порой и определяющий вклад в развитие геоинформационных технологий, представленных в данной работе, следует отметить Еськина Д.М., Рыбалова Н.Б., Вишнякова В.Ю, Ощепкова С.С., Степанова А.В, Панова А.С.

## ПРЕДСТАВЛЕНИЕ ИНЖЕНЕРНЫХ СЕТЕЙ

### 1.1. Типология инженерных сетей и классификация задач эксплуатации

Инженерные сети осуществляют централизованное снабжение рас­средоточенных потребителей электрической и тепловой энергией, топли­вом, водой или другими транспортируемыми продуктами. В настоящее время значение инженерных сетей в инфраструктуре городского хозяйст­ва продолжает возрастать. Спектр объектов, представляющих в сово­купности инженерную сеть, весьма широк и разнообразен. Кроме того, инженерные сети имеют четкую ведомственную принадлежность, кото­рая диктуется, главным образом, видом транспортируемого продукта, а также технологией его добычи, переработки, транспортировки и по­требления [1, 2]. Сложность вопросов проектирования и создания ин­женерных сетей определила тематику многих публикаций, научно-методических и алгоритмических разработок.

Инженерные сети могут быть типологизированы в зависимости от семантического и масштабного типа плана местности, физических и мате­матических признаков, а также технологических различий (рис. 1.1) [3, 4].

Первой, и наиболее общей, является классификация *по виду транс­портируемого продукта и способу транспортировки* [5]:

- *кабельные сети*: электрические воздушные, электрические ка­бельные подземные, низкого/высокого напряжения, контактные сети, телефонные сети; сети передачи данных (электрические, оптоволокон­ные), телерадиосети;

- *трубопроводные сети*: водоснабжение (горячее, холодное), во­доотведение (бытовая, ливневая, техническая канализация), теплоснаб­жение (с разными теплоносителями), газопроводные, нефтепроводные, продуктопроводные, вентиляционные, пневматические;

- *дорожные сети*: автомобильные и рельсовые дороги, метропо­литен, фуникулер и т. д.

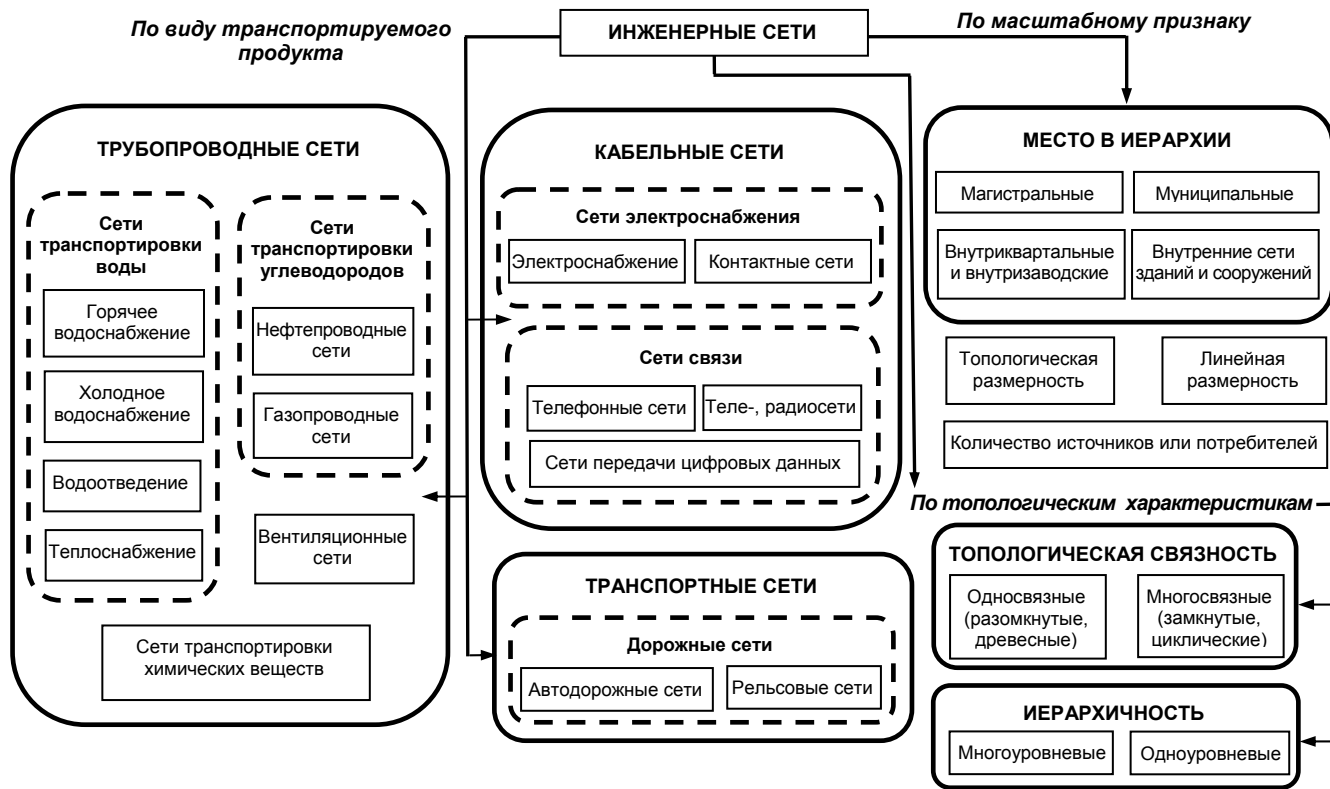


Рис. 1.1. Типология инженерных сетей

Достаточно значимой является классификация *по масштабному признаку* [4], в которой под масштабом подразумеваются следующие измерения: линейные размеры сети (длина линий), топологическая размерность (число узлов, линий), место в иерархии (магистральные, муниципальные, внутриквартальные и внутризаводские; внутренние сети зданий и сооружений).

Не менее важна и *топологическая структура сети* (топология сети при передаче продукта). Сети могут быть разомкнутыми (односвязными, древесными) и замкнутыми (многосвязными, циклическими).

Инженерные сети являются сложными объектами, имеющими протяженность не только в пространстве, но и во времени. Развитие инженерных сетей во времени определяется **жизненным циклом инженерных сетей** (рис. 1.2).

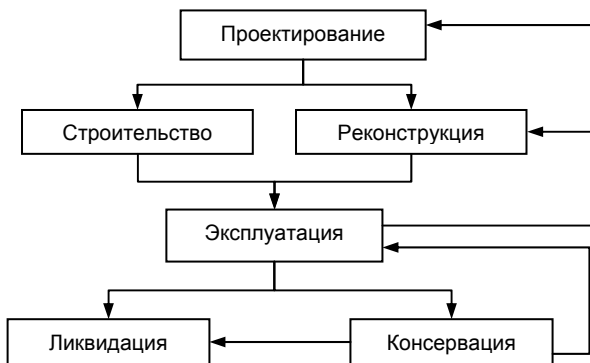


Рис. 1.2. Жизненный цикл инженерных сетей

Существование любой инженерной сети обусловлено необходимостью получения определенных услуг в той или иной хозяйственной отрасли, что ставит задачу создания инженерной сети определенного вида. Созданию инженерной сети предшествуют действия по **проектированию сети** (рис. 1.3) [4], которые предваряются процессом выявления будущих абонентов (потребителей услуг) сети. При этом выясняются приблизительные потребности абонентов (нагрузки) на основе типовых характеристик либо известных индивидуальных характеристик. При большом числе абонентов их обобщают по различным критериям: территориальной

близости, виду требуемой услуги. После **выявления потребностей** в услугах следует процесс **выбора источников услуг** в соответствии с суммарными потребностями абонентов. Для ряда сетей, в которых источники услуг заключены в самих сетях, этот процесс превращается в поиск расположения источников услуг.

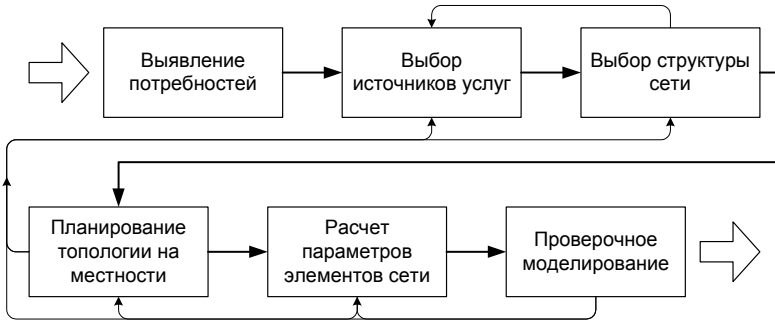


Рис. 1.3. Этап проектирования инженерной сети

Следующим процессом является **выбор структуры сети**. Здесь рассматривается создание циклических или разомкнутых сетей, их иерархичность. После выбора той или иной структуры или их совокупности производят **планирование топологии сети** с привлечением информации о районе расположения проектируемой сети, расположении абонентов и других сетей, особенностях рельефа местности и т. п.

Далее производится **расчет параметров элементов сети** в соответствии с заданными нагрузками и топологией сети. При этом выполняется расчет параметров оборудования, участков в нормальном режиме их функционирования. Результатом такого расчета является окончательная конфигурация сети.

После построения окончательной конфигурации сети производится **проверочное моделирование всей сети**, при котором исследуются режимы функционирования сети при различных внешних воздействиях.

На каждом процессе проектирования возможен переход на предыдущие процессы с целью коррекции структуры, топологии, используемых источников, параметров элементов сети. Этап проектирования завершается окончательным утверждением плана создания сети.



После проектирования сети следует этап физического воплощения проекта сети — **строительство**. Как правило, это длительный этап, совмещенный с эксплуатацией построенных частей сети.

Один из основных этапов жизненного цикла инженерной сети — **эксплуатация** (процесс предоставления услуг в соответствии с задачами, поставленными перед сетью на этапе проектирования). Эксплуатация инженерной сети начинается уже во время ее строительства. Этап эксплуатации можно разделить на некоторые процессы, происходящие одновременно в рамках одной и той же сети (рис. 1.4).

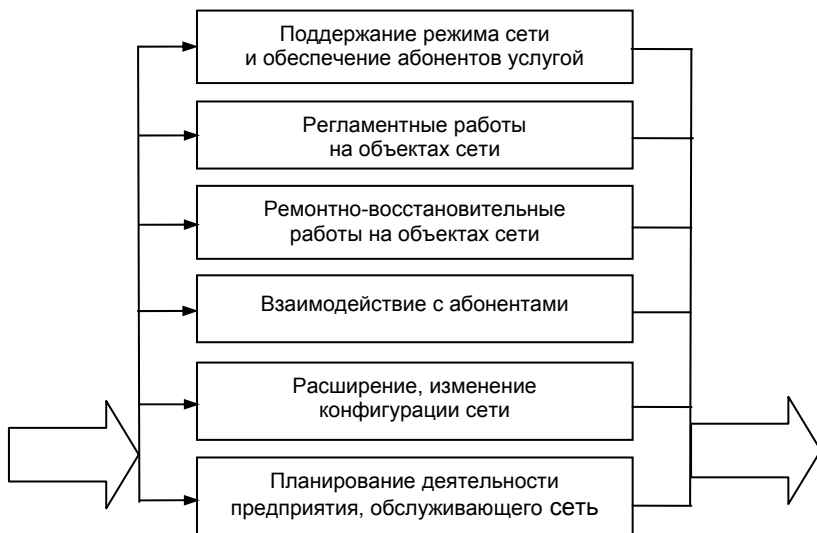


Рис. 1.4. Процессы этапа эксплуатации сети

Процесс *поддержания режима функционирования сети* должен осуществляться непрерывно, без перерыва в обеспечении абонентов услугой. Для этого требуется наличие средств управления режимом и средств контроля состояния сети. Задачи процесса решаются диспетчерскими службами, имеющими в распоряжении сведения о нагрузках, текущей топологии и конфигурации сети.

Проверка функциональности и надежности объектов сети осуществляется посредством проведения *регламентных работ* различного

содержания: осмотров, измерений режимных параметров, испытаний, мониторинга. Результатом таких работ могут стать выявленные неисправности или «узкие места».

Последствия аварий и обнаруженных неисправностей устраняются в процессе проведения *ремонтно-восстановительных работ*, при которых производится временное отключение части абонентов. *Взаимодействие с абонентами* заключается в информационной поддержке деятельности сети, извещении абонентов о плановых отключениях и т. п.

*Расширение и изменение конфигурации сети* производится без прерывания текущего снабжения абонентов и осуществляется либо за счет существующих сооружений сети, либо за счет строительства новых. Этот процесс можно рассматривать как проектирование или реконструкцию подсетей более мелкого масштаба, нежели основная сеть.

*Планирование деятельности предприятия*, обслуживающего сеть, необходимо для обеспечения его эффективной финансово-хозяйственной деятельности. Оно производится на основе агрегированных эксплуатационных данных, экономических и иных прогнозов.

В ходе эксплуатации сети могут быть выявлены новые потребности в услугах сети, обнаружено несовершенство исходного проекта, кроме того, обязательно отслеживание неизбежного процесса устаревания оборудования сети. Все это приводит к необходимости реконструкции сети. На **этапе реконструкции** осуществляется замена старых участков сети на новые либо всеобъемлющее изменение сети, сопровождающееся предварительным проектированием процессов реконструкции. Реконструкция, как правило, приводит к длительному перерыву в обслуживании абонентов. После реконструкции сеть снова переходит на этап эксплуатации.

Если потребность в услугах сети исчезает, то сеть может перейти на **этапы консервации** или **ликвидации сети**. Этап ликвидации сети завершает жизненный цикл сети. Этап консервации, в отличие от этапа ликвидации, не является последним в жизненном цикле. Сеть может быть законсервирована в случаях вероятности использования ее (или ее частей) в будущем либо при отсутствии материально-технических и/или финансовых возможностей для ликвидации сети.

Несмотря на значительные различия в типе транспортируемого продукта и технологиях, используемых эксплуатационными организациями, современные инженерные сети характеризуются рядом общих специфических особенностей [1]:

- 1) обширным территориальным распределением и огромным числом элементов, формирующих систему;
- 2) непрерывным развитием в пространстве и во времени;
- 3) сложной иерархической структурой управляемой и управляющих систем и непосредственным наличием субъекта в контуре управления;
- 4) непрерывностью во времени процессов транспортировки и распределения целевого продукта;
- 5) высокой степенью централизации управления с одновременной децентрализацией оперативного управления технологическими процессами транспортировки и распределения целевого продукта;
- 6) инерционностью процессов транспортировки продукта и необходимостью создания его оперативных запасов в резервуарах и хранилищах.

Наличие общих специфических особенностей позволяет формировать обобщенные классы прикладных задач управления инженерными сетями (табл. 1.1) [6].

### **Задачи инвентаризации, паспортизации, учета**

Данный класс задач является базовым в управлении инженерными сетями. Все виды сетей имеют средства мониторинга объектов и оборудования с целью инвентаризации, паспортизации и учета. С учетом различных подходов к инвентаризации можно выделить некоторые характерные черты [6]. Во-первых, целью инвентаризации является точное определение наличия объектов, оборудования сетей и их уникальная идентификация. Во-вторых, цель инвентаризации состоит в выяснении состояния оборудования и степени его износа. **Инвентаризация** — непрерывный процесс, сопровождающий ввод в эксплуатацию новых объектов и оборудования, обследование эксплуатируемых и списание старых объектов и оборудования.

Эта задача тесно связана с **паспортизацией объектов и оборудования**. Паспорт объекта (оборудования) включает:

- типовые инвентарные данные (даты создания и ввода в эксплуатацию, балансовую стоимость, инвентарные и заводские номера, текущий износ);
- технологическую информацию об объекте (тип, ГОСТ, мощность, типовые значения эксплуатационных и иных параметров и т. п.);
- параметрическую информацию об объекте (текущие эксплуатационные параметры, которые могут быть отличны от типовых).

Таблица 1.1

Классификация прикладных задач управления инженерными сетями

Класс задач	Состав прикладных задач
Инвентаризация, паспортизация, учет	Инвентаризация фондов сетей. Паспортизация объектов и оборудования сетей. Учет расходов материалов и энергоресурсов
Пространственное моделирование сетей	Ведение технологических схем. Ведение карт и планов расположения сетей. Моделирование профилей сетей
Предпроектный анализ	Расчеты нагрузок потребителей. Расчеты календарных изменений нагрузки. Расчеты режимов работы источников
Моделирование жизненного цикла сетей	Отслеживание, архивирование технологических событий, данных о неисправностях. Отслеживание, архивирование параметров объектов. Ведение архива коммуникаций
Эксплуатационный анализ	Расчеты режимов потокораспределения. Технологические и топологические расчеты. Расчеты потерь в сетях. Расчеты оптимального потокораспределения
Информационное обеспечение процессов управления и анализа инженерных сетей	Пространственные и атрибутивные запросы. Генерация отчетов, справок. Генерация планов работ

Отдельной задачей является *учет расходования материалов и энергоресурсов* в процессе эксплуатационной деятельности инженерных сетей. Эта задача имеет тесную связь с моделированием жизненного цикла объектов и оборудования сетей.

### **Задачи пространственного моделирования сетей**

Инженерные сети представляют собой пространственные объекты, имеющие, как правило, достаточно крупные масштабы. Взаимное расположение элементов инженерных сетей, их пространственная структура являются весьма важными характеристиками. Для эксплуатации любой инженерной сети необходима полная и достоверная информация о расположении объектов и оборудования сети, которая содержится в формах представления пространственной информации о сетях [6]:

- *принципиальных (технологических) схемах сетей*, в которых указаны параметры элементов сетей, потребителей, источников, состояний коммутаторов. Данные схемы представляют собой немасштабное изображение структуры сети. Как правило, они могут иметь разные уровни детализации, представляющие ту или иную часть сети. В различных сетях для изображения объектов и оборудования используются разные типовые обозначения. Эта информация необходима для диспетчерских служб и отделов планирования конфигурации сети. Схемы являются источником данных для расчетных задач и задач анализа топологии;

- *планах расположения объектов и оборудования сетей* на местности. Содержащаяся в них информация необходима для эксплуатационных бригад, планирования строительства, согласования вскрышных и других работ, экологического анализа, анализа близости различных объектов на местности и объектов сетей, проведения различного вида расчетов. Карты и планы территорий, где проложена сеть, составляют топооснову, которая используется при разработке планов расположения объектов и оборудования сети. Топооснова должна иметь определенную точность, необходимую для привязки к ней объектов сети.

Содержащаяся в планах информация используется для решения *задачи построения профиля участков* инженерных сетей [7].

### **Задачи предпроектного анализа**

Задачи этого класса чаще всего ставятся при проведении реконструкции во время эксплуатации. К ним относятся расчеты нагрузок потребителей, календарных и часовых графиков их изменения, режимов работы источников [6]. Особенностью расчета нагрузок потребителей является невозможность их точного предварительного определения в ряде случаев. В таких ситуациях используются типовые оценки для разных типов потребителей. Необходимость расчета календарных и суточных графиков нагрузок вызвана сильной зависимостью потребления от времени года и суток.

### **Задачи моделирования жизненного цикла объектов и оборудования**

Задачи моделирования жизненного цикла оборудования можно разделить на три основных вида [6]:

1) *моделирование технологического состояния оборудования и событий*. Для всех видов объектов и оборудования требуется иметь полную информацию о следующих событиях:

- времени обнаружения и устранения неисправностей;
- времени проведения регламентных и ремонтно-восстановительных работ с указанием исполнителей данных работ;
- динамике изменения остаточного ресурса;

2) **отслеживание параметров оборудования**, состоящее в регистрации и накоплении значений основных параметров оборудования, позволяющих проводить ретроспективный анализ их изменения. Это дает возможность предупреждать выход оборудования из строя, оценивать и учитывать его индивидуальные особенности;

3) **отслеживание конфигурации сети** во времени, заключающееся в накоплении информации обо всех изменениях режима работы элементов сети. Данная информация позволяет проводить ретроспективные анализы работы диспетчеров и режимов сети.

### Эксплуатационный анализ

К задачам этого класса относятся задачи расчета потокораспределения и режима, при решении которых используются следующие определения [6]:

- **потокораспределение** — набор величин значений потоков транспортируемого продукта в линиях передачи и значений потенциалов (давления, напряжения, очереди на обработку) в узлах инженерной сети;
- **режим** — набор значений управляющих параметров и соответствующее им потокораспределение. Режим должен удовлетворять потребностей сети с точки зрения обеспечения требуемых параметров.

Большинство задач эксплуатационного анализа базируются на предварительном вычислении потокораспределения [1]. Задачами данного класса являются:

- **прямой расчет потокораспределения**, составляющий основу решения большинства задач. Входными данными в задаче являются параметры оборудования и объектов сети, состояния коммутаторов, значения нагрузок абонентов. Результатом расчета является потокораспределение целевого продукта внутри инженерной сети. Прямой расчет потокораспределения дает информацию о функционировании сети и качестве обслуживания абонентов услугой;

- **расчет режима**, в котором входными данными являются параметры оборудования и объектов сети, текущие состояния коммутаторов, значения нагрузок абонентов и список управляемых параметров оборудования, доступных для регулирования. Результатом расчета яв-

ляется режим. Расчет режима позволяет получить информацию о величине воздействия на сеть, необходимой для ввода в режим, удовлетворяющий поставленным эксплуатационным и абонентским требованиям;

– *расчет требуемых технологических параметров*, для которого входными данными являются параметры оборудования и объектов сети, текущие состояния коммутаторов, значения нагрузок абонентов и список технологических параметров оборудования, доступных для изменения. Результатом расчета будет являться режим и набор измененных технологических параметров. С помощью данного расчета могут быть определены параметры оборудования для вновь создаваемой или реконструируемой сети, а также решена режимная задача при изменении регулируемых параметров элементов сети;

– *расчет потерь*, в котором входными данными являются показатели потокораспределения. Результатом расчета является набор значений энергетических, тепловых, мощностных или иных потерь для всех элементов сети и для сети в целом. Расчет потерь предоставляет информацию об эффективности функционирования сети. Кроме того, он позволяет выявлять «узкие места» сети и проводить экономическое обоснование для изменения конфигурации или параметров оборудования;

– *поиск участков сети с заданной топологией*, для которого входными данными являются показатели конфигурации сети, а выходными — участки сети с заданной топологией (например, односвязные участки);

– *оптимизационные расчеты*, в которых входными данными являются параметры оборудования и объектов сети, текущие состояния коммутаторов, значения нагрузок абонентов. Оптимизация производится по потерям, расходуемой энергии либо другим критериям. Результатом является оптимальный режим и набор измененных технологических параметров. Оптимизационные расчеты предоставляют информацию по оптимальному конфигурированию сети: выбор режима, в котором оборудование действует с максимальной эффективностью; возможная замена оборудования для достижения оптимального результата.

### **Задачи информационного обеспечения процессов управления и анализа инженерных сетей**

К задачам информационного обеспечения относятся задачи, выполняемые непосредственно в автоматизированной системе управления территориально распределенными сетями [6]:

- *построение запросов к графическим (пространственным) и атрибутивным данным;*
- *генерация отчетов, справок с использованием атрибутивных и графических данных;*
- *генерация планов работ с использованием необходимых графических видов, композиций.*

Приведенная классификация задач позволяет сформировать приоритеты по степени необходимости их выполнения:

1) задача оперативного получения информации об инженерных сетях в любой части интересующей территории, включая здания и сооружения;

2) задача совместного представления различных инженерных сетей на едином плане (топооснове), реализуемая с использованием задачи систематизации на топографической основе информации об объектах сетей;

3) задачи полного отслеживания неисправностей, регламентных и ремонтно-восстановительных работ, а также контроля их сроков и качества выполнения;

4) задача ведения архива документов по всем объектам сети и эксплуатационным событиям, по результатам которой осуществляется сбор данных для решения задачи автоматизации формирования текущих документов и выходных форм отчетности;

5) задачи информационного обеспечения для планирования работ по реконструкции и ремонту инженерных сетей, а также автоматическое отслеживание остаточного ресурса оборудования.

Вышеперечисленные задачи необходимо решать различным службам предприятий, эксплуатирующим или проектирующим соответствующий вид инженерных сетей. Сегодня невозможно себе представить, чтобы эти задачи решались без применения современных компьютерных технологий, что подразумевает создание автоматизированной системы управления инженерной сетью. Современным АСУ присущи, прежде всего, черты информационных систем, поскольку весомая доля их функций приходится на работу с базами данных паспортной и оперативной технологической информации.

Если говорить об уровне автоматизации отдельных классов рассматриваемых задач, то наиболее высоким он является для задач, связанных с паспортизацией объектов сети. Автоматизация деятельности



любого предприятия начинается, как правило, с решения проблемы перехода к безбумажной технологии ведения реестра объектов. Для автоматизации решения этой задачи применимы традиционные методы проектирования информационных систем [8].

Автоматизация прочих классов задач невозможна без создания сложной цифровой модели территориально распределенной инженерной сети. Компонентами цифровой модели должны быть: для задач представления сетей на картах и схемах — данные о координатной привязке и пространственном взаиморасположении объектов; для задач диспетчеризации и моделирования — данные о структуре сети и оперативная технологическая информация о ее параметрах; для задач оптимизации и анализа — данные о динамике изменения режимов сети, инфраструктуре и окружающей среде.

При создании информационных систем, решающих задачи управления инженерными сетями, объективно возникает ряд проблем, общих для разработки больших информационных систем со сложным программным обеспечением. Однако можно выделить проблемы, которые в той или иной мере можно считать специфическими, присущими только рассматриваемому классу информационных систем. Среди таких проблем основными являются следующие [9]:

1) информационная система должна выполнять большой блок геоинформационных функций, т. е. включать модули для визуализации и редактирования цифровых карт, выполнения пространственного анализа, поскольку решение задач, связанных с представлением сетей на топографических картах, топопланах и технологических схемах, требует хранения и оперирования пространственно привязанной информацией;

2) в информационной системе должна быть предусмотрена возможность установления между объектами сети различного рода отношений (топологических, ресурсных и др.), которые давали бы структурное определение сети, что обусловлено необходимостью хранения и оперирования информацией о структуре (топологии) сети для решения некоторых групп задач оперативного управления, оптимизации и др.

Автоматизированная информационная система, способная решать вышеперечисленные проблемы управления инженерными сетями того или иного типа, должна включать механизмы реализации формализованного математического описания модели данных ГИС с широкими возможностями по представлению неметрических отношений между

пространственными объектами, а также методы и алгоритмы решения прикладных задач управления инженерными сетями и инструментальные средства для создания и развития программного обеспечения проблемно ориентированных ГИС. Возможным решением описанных выше задач является использование **геоинформационных технологий** в процессе эксплуатации инженерной сети, предназначенных, прежде всего, для создания базы данных об истинных параметрах и топологии сети (ведение кадастра сети) [6].

В геоинформационной системе множеству реальных физических объектов сети соответствует множество информационных объектов, представляющих собой формальное описание реального объекта или субъекта социально-экономических отношений.

Описание информационного объекта включает:

- 1) идентификатор информационного объекта;
- 2) параметры территориальной привязки;
- 3) семантические характеристики (атрибуты).

Идентификатором информационного объекта является признак или набор признаков, уникальное значение которых позволяет выделить этот объект из множества других экземпляров объектов данного типа. Информация об экземплярах объектов различных типов хранится в системе. Идентификатор используется для однозначного определения экземпляра информационного объекта и организации информационного обмена в информационной системе предприятия.

Наличие территориальной привязки информационного объекта позволяет производить объединение данных об объекте в территориально административном разрезе. Территориальная привязка может задаваться в позиционном представлении или географическими координатами, а также в виде семантического признака.

Семантические характеристики (атрибуты) информационного объекта представляют собой совокупность признаков объекта. Состав атрибутов должен определяться с учетом их необходимости и достаточности для формального описания объекта при решении задач управления инженерной сетью.

Описание пространственных объектов в ГИС, т. е. формализованное представление их свойств, включает указание их позиционной и содержательной определенности. Позиционная часть описания данных организуется в определенные структуры *пространственных данных*,

связанных некоторыми отношениями с их *непозиционными данными* (атрибутами). Для того чтобы рассмотреть содержание любой инженерной сети, требуется указать возможные виды связей и взаимодействий, что достигается применением формализованного аппарата теории отношений.

## 1.2. Моделирование топологических отношений в инженерных сетях

Для решения широкого круга задач, связанных с управлением территориально распределенными сетями, необходимо, чтобы сеть была представлена как граф, т. е. помимо задания метрических характеристик объектов (координат местоположения, длины и пр.) требуется задание неметрических отношений между объектами, которые в геоинформатике традиционно называют топологическими [6].

Топология — раздел математики, изучающий топологические свойства фигур (т. е. свойства, не изменяющиеся при любых деформациях, производимых без разрывов и склеиваний, а точнее при взаимно-однозначных и непрерывных отображениях). Примерами топологических свойств фигур являются размерность, число кривых, ограничивающих данную область, и т. д. [9]. Топологические свойства фигур (называемые также топологическими инвариантами) отличают от метрических, которые обычно связывают с расстояниями между точками, углами между линиями, ребрами фигуры и т. д. [10].

Топологические характеристики не имеют смысла при наличии единственного объекта. О них говорят только при наличии нескольких объектов, системы объектов, так как они характеризуют взаимоотношения объектов. Поэтому данные характеристики часто называют топологическими отношениями. Наиболее часто встречающимися топологическими отношениями между пространственными объектами являются [8]:

- **связность** — наличие определенным образом установленной и поддерживаемой взаимосвязи между объектами;
- **соседство** (примыкание, близость) — наличие у объектов общей границы, точек;
- **вложенность** (включение) — один из связываемых объектов содержит внутри себя (в графическом и/или географическом смысле) другой;
- **пересечение** — один из связываемых объектов пересекает другой и наоборот.

В геоинформатике применение термина «топологический» не такое строгое, как в математике и связано с возможностью включения информации о топологических характеристиках пространственных объектов в модель данных конкретной ГИС [11].

В существующих ГИС используются различные способы организации описания реальности посредством использования моделей данных. Каждая модель имеет свою область применения, поэтому при необходимости решения большого числа задач следует использовать совокупность разных моделей [12]. Для лучшего понимания области представления данных в ГИС приведем несколько определений.

*Определение 1.* Представление пространственных данных, или **модель пространственных данных** — это способ цифрового описания пространственных объектов, тип структуры пространственных данных (способ структурного описания исходных данных) [6].

Выделяют четыре типа пространственных объектов: точечные (точки); линейные (линии); площадные (области, ареалы, полигоны); поверхности (рельефы). Объекты могут быть 0-, 1-, 2- и 3-мерные соответственно.

Модели пространственных данных могут иметь векторное или растровое представление, содержать или не содержать топологические характеристики, что позволяет классифицировать модели по трем типам [12]:

- 1) растровая модель;
- 2) векторная нетопологическая модель;
- 3) векторная топологическая модель.

Менее распространенной является гибридная модель, которая содержит характеристики как векторных, так и растровых моделей.

*Определение 2.* **Растровая модель данных** — это цифровое представление пространственных объектов в виде совокупности ячеек растра (пикселей) с присвоенными им значениями класса объектов. Растровое представление предполагает позиционирование объектов указанием их положения в соответствующей растру прямоугольной матрице единообразно для всех типов пространственных объектов (точек, линий, полигонов и поверхностей) [13].

*Определение 3.* **Векторная модель данных** — это цифровое представление точечных, линейных и полигональных пространственных объектов в виде набора координатных пар модели [14]. Если помимо описания геометрии объекта путем перечисления координатных пар узлов

в модели присутствует описание некоторых топологических характеристик, то такая модель называется векторной топологической, иначе — векторной нетопологической [6].

Как векторные, так и растровые модели имеют ряд достоинств, определяющих сферу применения каждой из них. Например, растровые модели, в отличие от векторных, могут передавать полутона, применяющиеся при визуализации рельефов местности. К достоинствам векторных моделей можно отнести небольшой объем памяти, требуемой для хранения модели, и относительную простоту редактирования графического представления объектов [12, 13, 15].

При решении задач управления инженерными сетями наиболее предпочтительными оказываются векторные топологические модели, так как *«основу анализа сетей определяет исследование связей между объектами, что задается топологией, или топологическими свойствами векторной модели»* [12].

Для векторной модели характерным является послойный принцип организации пространственной информации. Суть его заключается в том, что многообразная информация о какой-либо территории организуется в виде упорядоченной совокупности тематических слоев [11]. В каждом слое хранится информация об однотипных объектах.

Существует три подхода к типизации пространственных объектов [8]:

1) тематический, в котором однотипными считаются объекты, относящиеся к одной теме (например, объект «река» и объект «озеро» относятся к одной теме «гидрография»);

2) графический, в котором однотипными считаются объекты, относящиеся к одному классу векторных графических примитивов: точка, линия, полигон (в этом случае объект «река» и объект «озеро» не всегда относятся к одному типу, так как озеро всегда представляется в виде полигона, а река может представляться и в виде линии);

3) объектно-ориентированный, сочетающий в себе два предыдущих.

*Определение 4.* Совокупность тематических слоев, состоящих из объектов одного типа, описывающую единую область мирового пространства на основе векторной модели пространственных данных, называют **векторной цифровой картой** [8].

Послойная организация делает более гибким доступ к отдельным объектам и их тематическим совокупностям, а также упрощает пространственный анализ.

Существуют различные классификации топологических отношений, поддерживаемых современными ГИС [6]. При этом топологические отношения либо структурно заложены в векторную топологическую модель данных, используемую каждой конкретной ГИС, либо их поддержка обеспечивается дополнительно. Топологические отношения могут быть представлены отдельными видами [8].

**Необъектные топологии.** Малораспространенный подход, в котором картографическое изображение строится из топологически связанных примитивов (точек, линий и полигонов), каждый из которых не ассоциируется полностью с определенным пространственным объектом, а присутствует как общая составляющая сразу в нескольких объектах. Топологические отношения между примитивами строятся на основе линейно-узловой топологии, которая подробнее рассмотрена ниже.

**Объектные топологии.** В модели данных ГИС объектного типа, как правило, каждый пространственный объект ассоциирован с одним объектом модели (как графическим, так и атрибутивным).

Подавляющее большинство современных инструментальных ГИС поддерживают так называемую внутриобъектную топологию. Это означает, что топологические отношения связывают отдельные части, из которых строится каждый отдельный объект. Например, отдельные сегменты полилинии связаны между собой отношениями соседства и при редактировании такая конструкция стабильна — нельзя вырвать отдельный сегмент.

Из известных на сегодняшний день инструментальных ГИС внутриобъектную топологию поддерживают: MapInfo Professional (MapInfo Corp., США), AtlasGIS (ESRI, США), отечественная GeoCAD System (Новосибирск) и др.

Естественно, что внутриобъектной топологии недостаточно, если модель должна включать описание отношений между отдельными объектами. Такой вид топологических отношений получил название межобъектных топологических отношений, или межобъектной топологии.

В данном виде топологических отношений выделяют:

- узловые и линейно-узловые;
- межобъектные в пределах одного слоя;
- межслойные межобъектные;
- топологические межобъектные ресурсные связи;
- концептуальные топологические отношения.

*Линейно-узловые* и, более редкий вариант, *узловые* топологические отношения являются наиболее распространенным подвидом межобъектных топологий, поддерживаемых рядом ГИС.

Достоинствами линейно-узловой идеологии можно считать:

- экономию памяти – совместное хранение координат общих узловых точек для смежных объектов не порождает избыточность;
- возможное увеличение скорости отрисовки совокупности объектов в некоторых случаях за счет однократной прорисовки общих линейных сегментов (это преимущество используется не во всех ГИС, следующих этой идеологии).

К недостаткам линейно-узловой идеологии относятся:

- использование глобального списка точек, лишаящее объекты структурной автономности, т. е. невозможность изменения формы или положения объекта без вовлечения соседних объектов, что приводит к крайне затруднительной для пользователя процедуре редактирования карт;
- возможность связывания линейно-узловыми топологическими отношениями только полилинейных и полигональных объектов (без точечной или условно-точечной привязки);
- отсутствие информации о поддержке межслойных линейно-узловых топологических отношений базовыми структурами и функциями универсальных ГИС с такой идеологией;
- использование только одного типа топологических отношений — смежности (примыкания), отсутствие поддержки других видов (например, вложенности).

Яркими примерами ГИС, поддерживающих линейно-узловые топологические отношения, являются продукты фирмы ESRI (США) — Arc/Info, ArcView.

Некоторые развитые ГИС имеют возможность создавать сразу несколько видов *межобъектных топологий*. Наиболее ярким примером такой ГИС может служить AutoCAD Map (AutoDesk Inc., США). Этот пакет, являясь наследником мощных средств САПР, интегрированных с ГИС, предлагает развитые средства для построения моделей территориально распределенных инженерных сетей. В частности, AutoCAD Map поддерживает три типа топологии: узловую, сетевую и полигональную. Задав топологию в нескольких отдельных слоях, можно осуществлять операцию комбинирования этих топологий.

Как уже отмечалось ранее, одним из принципов цифрового картографирования является тематическое послойное деление. Особенностью территориально распределенных инженерных сетей является как большое (до нескольких десятков) число типов объектов сети, так и большое (до нескольких тысяч) число описываемых объектов. Следовательно, желательным было бы разнесение в разные слои объектов территориально распределенных инженерных сетей различных типов. Однако это желание порождает следующую проблему: лишь небольшое количество универсальных ГИС поддерживает *межслойные топологические* отношения, а не только отношения между объектами в пределах одного слоя. Например, ГИС AutoCAD Map позволяет задать топологию в пределах одного слоя, для межслойного пространственного топоанализа необходимо выполнение дополнительных процедур.

*Топологические ресурсные связи* — это топологические отношения, которые кроме обозначения пространственных топоотношений отражают расположение в пространстве реальных физических каналов передачи вещественных, энергетических и информационных ресурсов через границы объектов, отражаемых на цифровой карте [8]. Ресурсная связь, таким образом, расширяет понятие топологических отношений. Ресурсные связи позволяют выделить из всей совокупности пространственных отношений между объектами инженерных сетей (близости, включения и пр.) отношения, задающие топологию сети для моделирования физических процессов распределения ресурсов.

Также существует понятие *концептуальных топологических отношений* [8], позволяющих устанавливать топологические отношения между целыми классами связываемых объектов. Данное понятие вводится непосредственно для так называемых объектно-ориентированных ГИС, которые используют парадигму объектно-ориентированного проектирования (ООП) как для структурной организации самой инструментальной ГИС, так и для описания предметной области и построения прикладной ГИС.

В качестве некоторых правил выделения концептуальных топологических отношений, как правило, вводят следующие:

- объекты класса «здания» не должны пересекать границы объектов класса «земельные участки»;
- объекты класса «здания» не могут накладываться на объекты классов «дорога» или «река»;



– изображенная на карте энергетическая подстанция из соответствующего класса должна быть обязательно подключена к соответствующей кабельной сети и др.

Концептуальные топологические отношения, установленные для классов объектов, предполагают автоматическое их установление и для экземпляров этих классов.

Проведенный авторами анализ современных универсальных ГИС показывает, что модели данных ни одной из таких систем не обладают достаточной полнотой и гибкостью для представления данных об объектах территориально распределенных инженерных сетей. Описание модели данных должно быть определенным образом формализовано, т. е. опираться на уже существующие или разрабатываемые математические аппараты, в первую очередь в части формализации топологических отношений.

Общим недостатком подходов к формализации топологических отношений можно считать излишнее стремление к универсальности, которое приводит к слабой применимости предложенных моделей непосредственно при разработке модели данных, описывающей территориально распределенные инженерные сети. Использование нераспространенных моделирующих аппаратов делает модель закрытой для широкого круга разработчиков информационных систем.

В связи с этим актуальным представляется создание формальной векторной топологической модели данных на основе более доступных средств математики, например на основе аппарата теории множеств. Для управления территориально распределенной инженерной сетью модель данных инженерной сети должна быть максимально абстрагирована от частных особенностей конкретных видов сетей и иметь возможность легкой настройки и адаптации при создании той или иной проблемно-ориентированной системы управления инженерными сетями выбранного типа.

Авторами предлагаются принципы построения моделей территориально распределенной сети.

**Первый принцип:** модель данных территориально распределенной инженерной сети должна включать средства описания не только технологических объектов (их пространственных характеристик и атрибутов), но и описания предметной области в целом (описания классов объектов, атрибутов классов, назначение объектов в ресурсной сети и пр.).

Следовательно, разрабатываемая модель данных должна являться результатом интеграции двух составляющих: метаданных (описания предметной области в целом) и предметной составляющей (значения атрибутов конкретных объектов инженерных сетей).

**Второй принцип:** модель данных должна включать в себя средства представления как пространственных, так и атрибутивных характеристик объектов инженерных сетей, причем в модели должна присутствовать возможность одновременного представления пространственных характеристик объектов инженерных сетей на нескольких разномасштабных картах и технологических схемах.

Отличие векторных топологических моделей пространственных данных от нетопологических состоит в наличии специальных средств описания топологических отношений между пространственными объектами. Однако, как показывает проведенный анализ, учет только топологических отношений недостаточен при представлении структуры инженерной сети.

**Третий принцип:** модель данных территориально распределенной инженерной сети должна описывать различные типы отношений между пространственными объектами. Поэтому предлагается подход к представлению в моделях данных ГИС комплекса отношений различных типов между пространственными объектами.

В рамках решаемой проблемы управления инженерными сетями авторами данного исследования предлагается выделить следующие типы отношений между объектами:

- **топологические**, вытекающие из взаиморасположения объектов в топологическом пространстве, в частности на местности;
- **ресурсные**, отражающие связь объектов между собой в части выполнения функций по транспортировке какого-либо продукта (ресурса);
- **иерархические**, указывающие на принадлежность объектов к отношениям типа «хозяин–подчиненный», «родитель–потомок», «целое–часть» и т. д.;
- **логические** — прочие виды отношений, определяющих взаимное влияние объектов в рамках модели.

Как показывает практика, пространственные объекты инженерных сетей (особенно технологические) легко поддаются типизации: выделяются подмножества объектов, выполняющих одинаковые функции и обладающих одинаковым набором свойств, которые нужно учитывать

при решении ряда прикладных задач. При этом отмечается, что объекты одного типа вступают, как правило, в одни и те же отношения с объектами других типов и между собой. Следовательно, можно выделить в отдельную группу отношения между типами (классами) пространственных объектов (концептуальные отношения).

Любая пара типов или экземпляров объектов может находиться в отношениях более чем одного типа одновременно. В этом случае предлагается рассматривать композицию отношений.

**Четвертый принцип:** модель должна давать возможность описания сразу нескольких территориально распределенных инженерных сетей с разными транспортируемыми ресурсами.

**Пятый принцип:** для формализованного описания модели данных инженерных сетей следует использовать выбранный в результате анализа математический аппарат теории множеств; при этом одни составляющие модели будем представлять как базовые множества, а другие — вводить через базовые операции над множествами, такие как объединение, пересечение, прямое произведение и т. д.

## Глава 2

# МОДЕЛИРОВАНИЕ ИНЖЕНЕРНЫХ СЕТЕЙ

### 2.1. Алгебраическая модель инженерной сети

#### Плоская алгебраическая модель инженерной сети

Формально инженерная сеть  $S$  описывается в работах [16–18] следующим образом:

$$S = \{W^O, C, A(W^O), F(W^O), D, H\},$$

где  $W^O$  — множество объектов, образующих инженерную сеть;

$C$  — множество технико-экономических характеристик сети;

$A(W^O)$  — множество состояний объектов;

$F(W^O)$  — множество функций, выполняемых объектами инженерной сети;

$D$  — множество пользователей (подразделений), эксплуатирующих объекты сети;

$H$  — множество потребителей конечного продукта.

Основной функцией инженерной сети является обеспечение множества потребителей конечным продуктом, т. е. приведение объектов  $W^O$  в необходимое состояние  $A(W^O)$ , при котором они должны выполнять множество функций  $F(W^O)$  под руководством эксплуатационных служб.

Инженерная сеть может разделяться на множество эксплуатационных участков сети  $S = (S_1 \cup S_2 \cup \dots \cup S_N)$ ,  $S_i = \{W_i^O, C_i, A(W_i^O), F(W_i^O), D_i, H_i\}$ , где  $i$  — номер эксплуатационного участка. Данное разбиение позволяет повысить эффективность управления за счет обозримости каждого отдельного участка путем выделения отдельных элементов сети по принадлежности к отдельным этапам технологического процесса.

Процесс транспортировки продукта по функционирующей сети может быть описан вектором (направлением), что позволяет использовать ориентированный граф (множество точек, соединенных между собой направленными отрезками) для представления каждого уровня сети. Такое описание является общепринятым в мировой и отечественной практике.

Как правило, в литературе графы обозначаются с помощью символической формы  $W = \{U, G\}$ , где  $U$  — множество вершин графа,  $G$  — множество ребер графа. Множество вершин  $U$  графа представлено множеством некоторых элементов  $u$  ( $u \in U$ ), причем каждый элемент  $u_i$  имеет отношения с одним или несколькими элементами другого множества  $G$ . Данное формальное описание инженерной сети с использованием теории множеств и теории графов будем называть **плоской моделью территориально распределенных инженерных сетей**. В инженерных сетях независимо от их назначения можно выделить ряд общих с точки зрения топологии элементов (объектов) (рис. 2.1) [6]:

1) *источник* — узловой элемент. В электроснабжении это может быть источник напряжения, трансформаторная подстанция; в водоснабжении — водонапорная башня, скважина; в теплоснабжении — котельная, ТЭЦ. Источник может иметь два состояния: «включен» или «отключен»;

2) *потребитель* — узловой элемент. Это потребители воды, газа, электрической и тепловой энергии. Потребитель может иметь два состояния: «подключен» или «отключен»;

3) *отсекающее устройство (коммутатор)* — узловой элемент. В трубопроводных сетях — это запорная арматура (вентили, задвижки, краны); в электроснабжении — рубильники, выключатели, контакторы. Коммутатор может иметь два состояния: «открыто» или «закрыто»;

4) *простой узел*, служащий для соединения участков и всегда имеющий одно состояние — «открыто»;

5) *участок* — линейный объект, соединяющий пару узлов. Это кабели, ЛЭП, участки трубопроводов. В зависимости от конкретной реализации участок может иметь состояния: «открыт» или «закрыт». Кроме того, участок имеет направление от начального узла к конечному узлу.

Следует отметить, что в реальной сети коммутационное оборудование обычно расположено в узлах сети и образует в совокупности с ними *узлы распределения*. Кроме того, имеется еще один тип узлов — *узлы преобразования*, в которых происходит изменение рабочих параметров транспортируемого продукта, (трансформаторы, насосы, регуляторы давления и т. п.).

Создаваемая для расчетов сеть является моделью, а не полной копией реальной сети. Модели имеют ряд упрощений, суть которых состоит в однолинейном представлении участков, изменении степени детализации, необходимой настройки точности и подробности изображения.

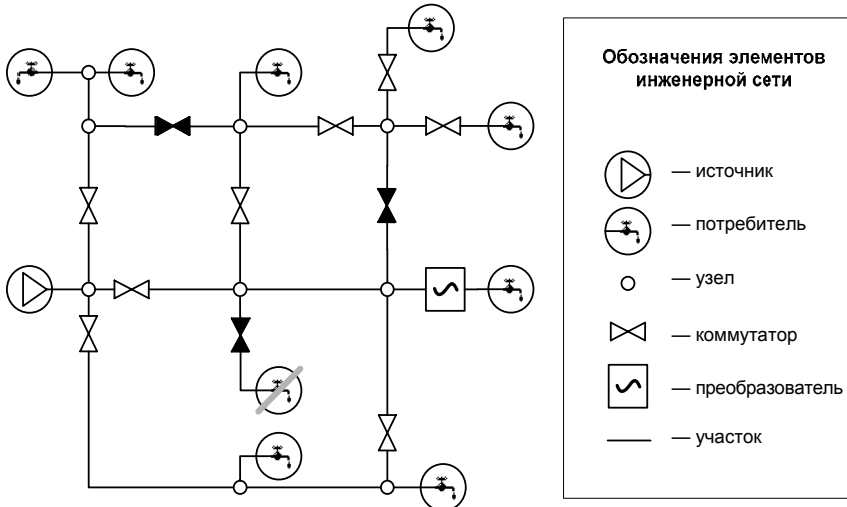


Рис. 2.1. Пример внутренней структуры плоской инженерной сети

В некоторых сетях отдельные фрагменты содержат несколько параллельно идущих участков. Для модели не является обязательным представление сети в виде отдельных элементов. Пользователь задает представление участков сети одним объектом, а расчетная задача, если это необходимо, интерпретирует внешнее представление сети необходимыми элементами. Таким образом, для расчетной модели инженерной сети возможно *однолинейное представление участков*.

*Степень детализации при изображении сети* может быть разной в зависимости от требований модели. Например, в водопроводной сети могут присутствовать сотни задвижек, назначение которых — перекрывать те или иные участки сети. Модель может быть построена так, что изображение задвижек будет необязательным. Вместо задвижки можно просто включать/отключать сам участок (рис. 2.2), а физическое влияние задвижки отразить в атрибутах с помощью коэффициента местного сопротивления. Показанные на рис. 2.2 схемы эквивалентны, но правая схема содержит на три узла и три участка меньше по сравнению с их числом на левой схеме. Когда таких «лишних» объектов тысячи и для них нужно заносить десятки атрибутов, время ввода существенно замедляется.

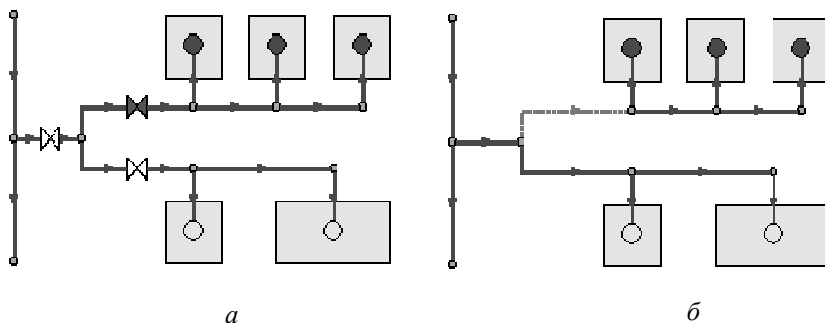


Рис. 2.2. Сравнение различных видов представления инженерных сетей:  
*a* — начальная схема сети; *б* — упрощенная схема сети

Геодезическая точность задания координат и обязательное наличие всех точек переломов на участках в некоторых расчетных задачах не имеют большого значения. Например, повороты и изгибы проводника никак не влияют на силу протекающего в нем тока. Важна общая длина провода, которую можно задать как атрибут. С одной стороны, очень удобно, когда расчетный граф сети привязан к местности, но, с другой стороны, ввод упрощенной схемы сети позволяет инженерам быстро начать расчеты. Таким образом, *точность и подробность изображения* не являются обязательными условиями при построении расчетной модели инженерной сети.

На рис. 2.3 изображены два способа задания одного и того же участка тепловой сети. Верхний участок соединяет узлы прямой линией. Нижний участок соединяет те же узлы, но линия выполнена с прорисовкой П-образных компенсаторов, которые по определенным законам влияют на гидравлическое сопротивление сети.

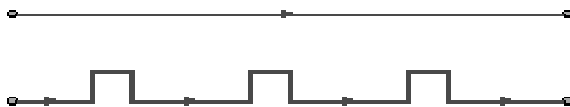


Рис. 2.3. Два способа задания одного и того же участка сети

По изложенным выше причинам множество объектов паспортизации и множество объектов расчетной модели одной и той же сети не являются полностью идентичными. Например, один расчетный участок может состоять из нескольких эксплуатационных. Попытка использовать инвентаризационные объекты в качестве объектов расчетной модели приводит к ее значительному и неоправданному усложнению. С одной стороны, достаточно удобно, когда расчетный граф сети привязан к местности. С другой стороны, качественный в плане геодезической привязки и полный в плане паспортизации ввод элементов сети существенно отдалает момент начала расчетов, ничего не добавляя к ним содержательно. Поэтому во многих организациях схема сети для паспортизации и технологическая схема для расчетов ведутся параллельно, хотя это и требует последующего согласования нескольких представлений одной и той же сети.

### **Многоуровневая модель инженерной сети**

Приведенная плоская модель территориально распределенных инженерных сетей может быть расширена при использовании различных иерархических уровней представления ее объектов. Пусть каждая вершина на верхнего уровня содержит подсеть следующего уровня  $u_{ij} = \{U_{ij}, G_{ij}\}$ , где  $i$  — номер уровня,  $j$  — номер вершины на  $i$ -м уровне. Например, в сети городского водоснабжения участками являются водозабор, насосно-фильтровальная станция, городская распределительная сеть, магистральные водоводы и т. д. В свою очередь, городская распределительная сеть делится на участки, питающиеся от разных подъемных насосных станций.

Таким образом, множество объектов сети можно представить трехуровневой структурой (рис. 2.4) [1]. Объектами первого уровня  $W_1^O$  являются эксплуатационные участки; объектами второго уровня  $W_2^O$  — сооружения, колодцы и прокладки; объектами третьего уровня  $W_3^O$  — внутренние схемы колодцев и сооружений, состоящие из оборудования (насосов, обратных клапанов, задвижек) и соединяющих их труб.

В сети водоснабжения на первом уровне вершинами множества  $U_1$  являются водозабор, насосные станции, городская сеть, а ребрами  $G_1$  — магистральные водоводы. На втором уровне —  $U_2 = \{C, K\}$ , где  $C$  — множество сооружений,  $K$  — множество колодцев;  $G_2$  — множество прокладок.



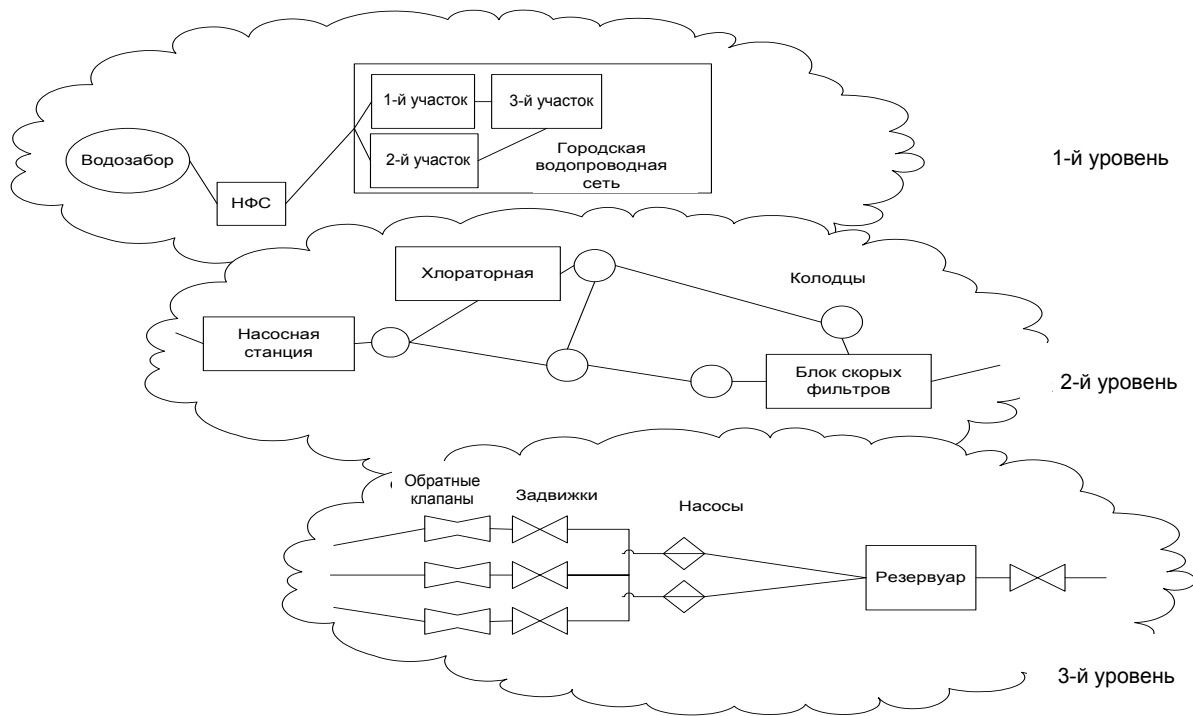


Рис. 2.4. Пример трехуровневого представления инженерной сети

Третий уровень в иерархической структуре сети водоснабжения представляет большой интерес, так как непосредственно на нем решаются задачи инвентаризации и оптимального управления (для водопроводных систем это гидравлические расчеты). Вершинами множества  $U_3$  будут являться источники давления, потребители (точки расхода) и места пересечения труб. Множество ребер  $G_3$  представляется участками труб, насосами, задвижками, регуляторами расхода и давления.

Помимо направленности процессов транспортировки в инженерных сетях есть и второе важное свойство — однозначная топологическая привязка: для каждого элемента сети можно указать соответствующую составляющую процесса транспортировки (поступление продукта, потребление, выдача или вообще отсутствие информации). На основе этих двух свойств картину интерпретации потоков в инженерной сети можно представить, как показано на рис. 2.5.

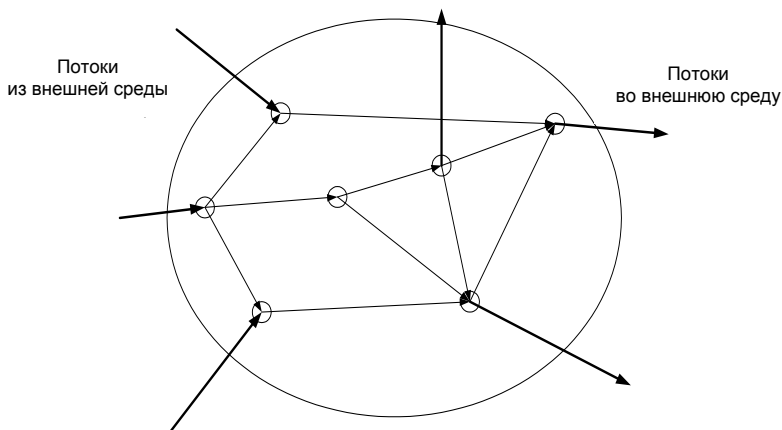


Рис. 2.5. Картина потоков в инженерной сети

На основе вышеописанного междууровневые отношения в инженерной сети можно представить в виде  $W_i = \bigcup_{j=1}^n W_{i-1}$ , где  $i$  — номер текущего уровня сети;  $j$  — номер подсети нижнего уровня;  $n$  — число подсетей на нижнем уровне.

### Взаимодействие объектов на различных уровнях инженерной сети

Правила выполнения различных действий над графами предполагают в качестве основы применение такого понятия, как «отображение множеств». Пусть  $g_i = \Gamma u_i$  — оператор отображения. Тогда граф можно записать в виде  $W = \{U, \Gamma u\}$ , где  $W$  считается заданным, если задано непустое множество  $U$  и отображение  $\Gamma u$ .

Правила выполнения действий с инженерными сетями будут основываться на действиях над графами («традиционных» операциях над множествами: объединении сетей, пересечении сетей, разности сетей) и специальных действиях, таких как выборка сетей, сечение сети, разрез сети, вставка сети и стягивание сети, учитывающих особенности построения структуры инженерных сетей [18].

**Объединение сетей:**  $W(U, \Gamma u) = W_1(U_1, \Gamma_1 u) \cup W_2(U_2, \Gamma_2 u)$ , где  $U = U_1 \cup U_2$ ,  $\Gamma u = \Gamma_1 u \cup \Gamma_2 u \cup \Gamma_{12} u$ , а  $\Gamma_{12} u$  — множество операторов отображения, объединяющих множества вершин. На рис. 2.6 приведен пример объединения сетей (рис. 2.6, а и б). Результат действия объединения (рис. 2.6, в) получается путем задания объединяющего множества вершин, указания существующих в исходных сетях связей и введения новых связей, в частности отображений  $(U_2, U_6)$  и  $(U_3, U_5)$ . Данное действие необходимо выполнять при включении новых участков в инженерную сеть.

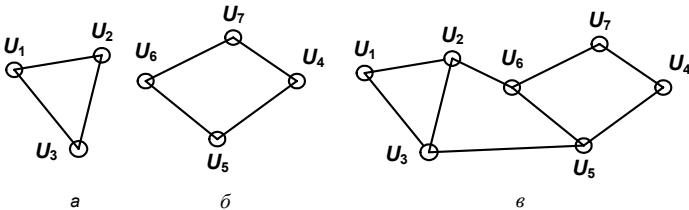


Рис. 2.6. Объединение сетей: а, б — первая сеть  $W_1(U_1, \Gamma_1 u)$  и вторая сеть  $W_2(U_2, \Gamma_2 u)$  операции объединения; в — результат  $W(U, \Gamma u)$  операции объединения первой и второй сетей

**Пересечение сетей:**  $W(U, \Gamma u) = W_1(U_1, \Gamma_1 u) \cap W_2(U_2, \Gamma_2 u)$ , где  $U = U_1 \cap U_2$ ,  $\Gamma u = \Gamma_1 u \cap \Gamma_2 u$ . При выполнении операции пересечения видно, что общих вершин у рассматриваемых сетей, представленных на рис. 2.6, нет.

Поэтому обратимся к другому варианту, представленному на рис. 2.7. Две сети (рис. 2.7, а и б) имеют как общие, так и различные вершины и отображения. На общих вершинах и отображениях образуется сеть пересечения (рис. 2.7, в).

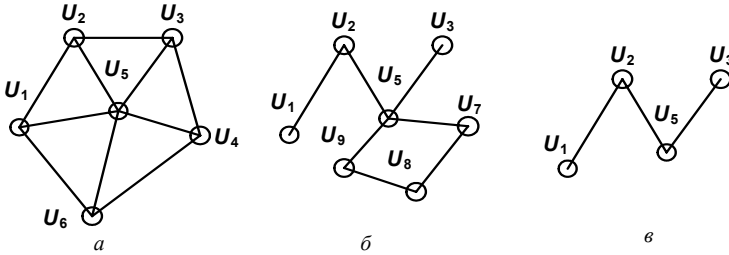


Рис. 2.7. Пересечение сетей: а, б — первая сеть  $W_1(U_1, \Gamma_1 u)$  и вторая сеть  $W_2(U_2, \Gamma_2 u)$  операции пересечения; в — результат  $W(U, \Gamma u)$  операции пересечения первой и второй сетей

**Разность сетей:**  $W(U, \Gamma u) = W_1(U_1, \Gamma_1 u) \setminus W_2(U_2, \Gamma_2 u)$ . Вершины сети разности  $W(U, \Gamma u)$  (рис. 2.8, в) сетей, представленных на рис. 2.8, а и б, находятся как разность множеств  $U_1$  и  $U_2$ :  $U = U_1 \setminus U_2$ , т. е. разность сетей представляет все вершины сети  $W_1$  за исключением общих вершин с сетью  $W_2$ .

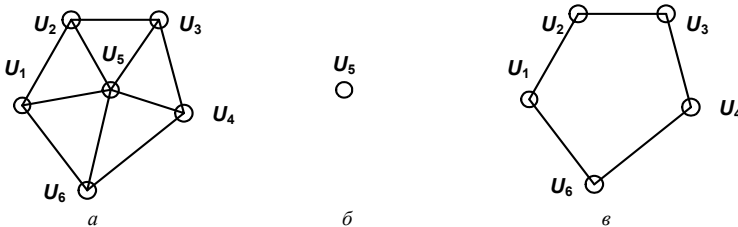


Рис. 2.8. Разность сетей: а, б — первая сеть  $W_1(U_1, \Gamma_1 u)$  и вторая сеть  $W_2(U_2, \Gamma_2 u)$  операции разности; в — результат  $W(U, \Gamma u)$  операции разности

Отображением для каждой вершины сети  $W(U, \Gamma u)$  является разность между множеством отображаемых вершин  $W_1(U_1, \Gamma_1 u)$  и отображением рассматриваемой вершины в графе, т. е.  $\Gamma u_i = \Gamma_1 u \setminus \Gamma_1 u_i$  (см. рис. 2.8).

По аналогии со специальными реляционными операциями, определенными Коддом, введем специальные сетевые действия (выборку и сечение сетей, разрез, вставку и стягивание сети), модифицированные с учетом того, что их операндами являются участки сетей, а не отношения.

**Выборка сетей:**  $W(U, \Gamma u) = \text{Choice}(W_1(U_1, \Gamma_1 u), \text{Conditions})$ . Операция  $\text{Choice}()$  возвращает подсеть, содержащую все узлы и ребра, удовлетворяющие определенным условиям  $\text{Conditions}$ , в сеть  $W(U, \Gamma u)$ . Пример операции выборки приведен на рис. 2.9. На сеть  $W_1(U_1, \Gamma_1 u)$ , изображенную на рис. 2.9, *а*, накладываются различные условия выборки:

- 1)  $\text{Conditions} =$  «Выбрать все элементы подсети, принадлежащие первому кварталу» (рис. 2.9, *б*);
- 2)  $\text{Conditions} =$  «Выбрать все элементы подсети, расположенные на улице ...» (рис. 2.9, *в*);
- 3) если  $\text{Conditions} = 0$ , то  $W(U, \Gamma u) = W_1(U_1, \Gamma_1 u)$ .

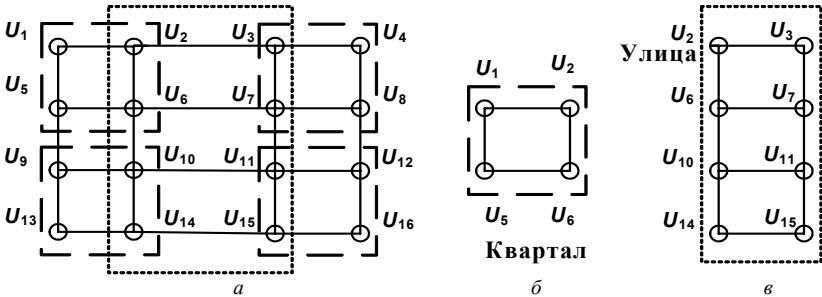


Рис. 2.9. Выборка сети: *а* — сеть  $W_1(U_1, \Gamma_1 u)$ , на которой необходимо выполнить операцию выборки; *б* — результат  $W(U, \Gamma u)$  операции выборки для условия (1); *в* — результат  $W(U, \Gamma u)$  операции выборки для условия (2)

**Сечение сети:**  $W(U, \Gamma u) = \text{Section}(W_1(U_1, \Gamma_1 u), \Gamma_0 u)$ . Результат операции  $\text{Section}$  — это разделяющее множество, составленное из минимальной совокупности ребер, при удалении которых нарушается его связанность. На рис. 2.10 представлен пример сечения сети (рис. 2.10, *а*) и разделяющего ее множества ребер (рис. 2.10, *б*). Сечение задается перечнем входящих в него ребер, число которых будем называть *рангом сечения*. Как правило, необходимо рассматривать сечение инженерной сети, заданное относительно источников сети для определения зоны их влияния.

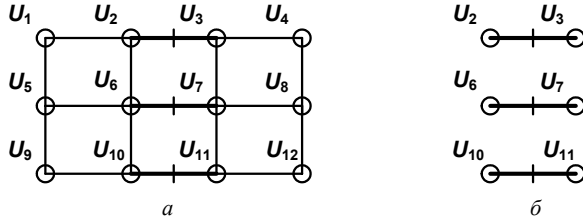


Рис. 2.10. Сечение сети: *a* — сеть  $W_1(U_1, \Gamma_{1u})$ , на которой необходимо выполнить операцию выборки; *б* — разделяющее множество ребер  $\Gamma_{0u}$

**Разрез сети:**  $W(U, \Gamma_u) = \text{Cut}(W_1(U_1, \Gamma_{1u}))$ . Для изменения структуры и связанности сети из нее удаляются отдельные вершины и связанные с ними исходящие или входящие дуги.

Пусть рассматриваются пути между вершинами  $U_5$  и  $U_8$  (рис. 2.11). Зададим на графе  $W_1(U_1, \Gamma_{1u})$  подмножество вершин  $L$ , чтобы вершина  $U_5 \notin L, U_8 \in L$ . Тогда разрез (5–8) образуется при удалении вершин, принадлежащих подмножеству  $L$ , в которые заходят дуги, рассекающие в совокупности все пути (5–8).

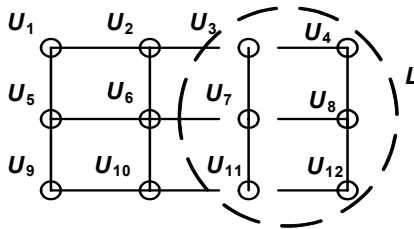


Рис. 2.11. Разрез сети

**Вставка сети:**  $W(U, \Gamma_u) = W_2(U_2, \Gamma_{2u}) \subset W_1(U_1, \Gamma_{1u})$ . Сеть  $W(U, \Gamma_u)$  — это вставка подсети нижнего уровня в сеть верхнего уровня (рис. 2.12).

Вершина  $U_0$  верхнего уровня (рис. 2.12, *a*) заменяется множеством вершин и ребер нижнего уровня (рис. 2.12, *б*). Результирующая сеть показана на рис. 2.12, *в*.

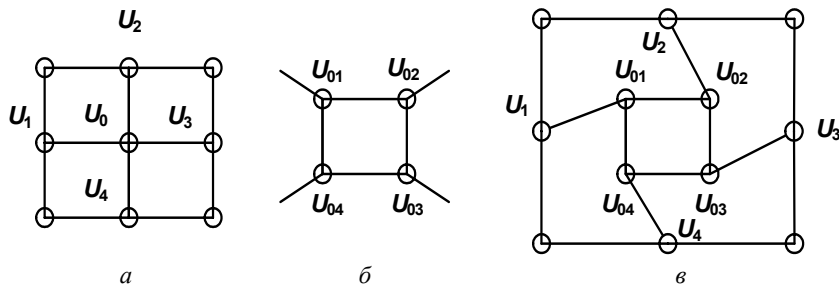


Рис. 2.12. Вставка сети: *a* — сеть верхнего уровня  $W_1(U_1, \Gamma_1 u)$ ; *б* — подсеть нижнего уровня  $W_2(U_2, \Gamma_2 u)$ ; *в* — результат  $W(U, \Gamma u)$  операции вставки подсети нижнего уровня в сеть верхнего уровня

**Стягивание сети:**  $W(U, \Gamma u) = W_2(U_2, \Gamma_2 u) \not\subset W_1(U_1, \Gamma_1 u)$ . Замена подсети нижнего уровня одним узлом в сети верхнего уровня (операция обратная операции вставки).

Вышеописанные операции будем называть операциями сетевой алгебры. Под алгеброй, как правило, понимают математический аппарат для формального структурированного описания алгоритмов функционирования систем. Сетевая алгебра — математический аппарат для манипулирования сетевой моделью данных с операциями, приведенными выше.

Данные операции находят практическое применение в задачах определения возможностей взаимосвязи двух узлов сети в случае выхода из строя некоторых промежуточных узлов, когда узел и участки, примыкающие к нему, оказываются неработоспособными (например, при аварии) и становится невозможным прием или транспортировка целого продукта к другим узлам. В теоретическом отношении эти ситуации соответствуют концепциям описанных выше процедур и позволяют оценить принципиальные возможности реализации процессов транспортировки продукта в отдельных участках сети в различных условиях (отслеживание и обработка (моделирование) переключений), а также рассчитать значения характеристик потоков (выполнение гидравлических и энергетических расчетов) и обеспечить тематический анализ схемы.

## 2.2. Унификация параметров и функций алгебраической модели инженерной сети

### Общие положения по разделению конкретных сущностей по классам абстракций

Для качественного разделения объектов по классам и наиболее простого и в то же время полного описания, удобного для анализа, необходимо выработать единую методику абстракции конкретных сущностей реального мира инженерных сетей на основе унификации их параметров и функций. Общие положения по разделению конкретных сущностей по классам состоят в следующем. На верхнем уровне инженерной сети выделяется абстрактный объект, для которого определяются присущие множеству объектов сети общие характеристики: уникальное имя объекта, его тип (принадлежность к тому или иному виду сущностей), состояние. Набор возможных состояний будет расширяться по мере конкретизации описания объекта. Объект характеризуется списком подобъектов, находящихся в его составе. Функции выделенного абстрактного объекта (изменение состояния объекта, добавление нового объекта и исключение определенного объекта из состава) могут рассматриваться как общие для множества объектов, входящих в сеть (рис. 2.13)<sup>1</sup>.

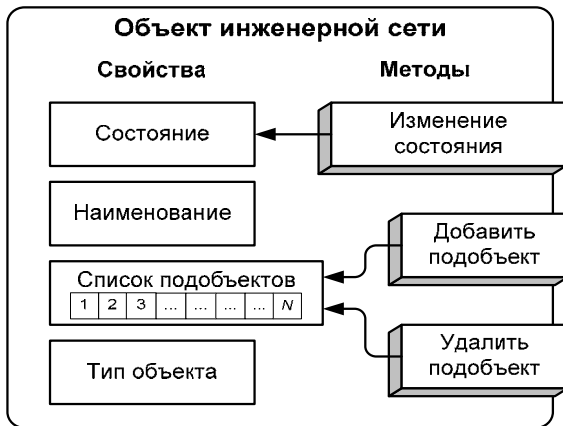


Рис. 2.13. Представление абстрактного объекта инженерной сети

<sup>1</sup> Рисунки 2.13–2.23 выполнены по материалам работы Сарычева Д.С. [4].



Следующий уровень абстракции представлен такими объектами, как оборудование, строительные сооружения, линии, производственно-технологические участки. Поскольку каждый вид сетей на данном уровне может быть конкретно детализирован, объекты данного уровня описываются как «специфические».

Общими характеристиками для любого вида *абстрактного оборудования* могут быть следующие:

- 1) обязательное инвентарно-производственное описание;
- 2) некоторый количественный показатель износа оборудования и балансовая стоимость;
- 3) тип оборудования (это перекрывает тип объекта), марка и конкретное исполнение;
- 4) исполняемая технологическая функция.

К стандартным методам объекта добавляются формирование инвентарного отчета, выдача информации о типовых (справочных) параметрах оборудования данного типа, марке и исполнении. Данное описание предназначено для целей инвентаризации, получения интегральных экономических оценок, а также расчетно-аналитических целей (рис. 2.14).

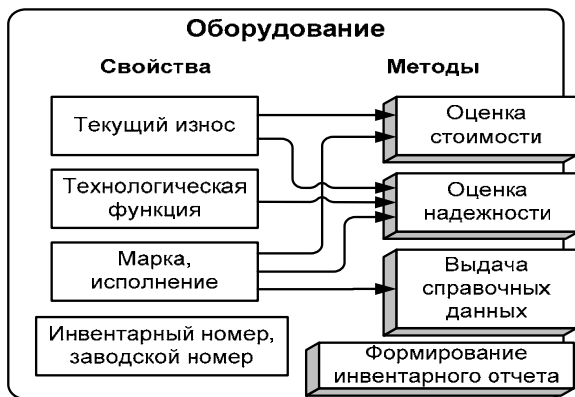


Рис. 2.14. Представление абстрактного оборудования инженерной сети

*Строительное сооружение* может выполнять разные роли: несущую, защитную, технологическую. Строительное сооружение имеет ряд технологических отличий от прочих объектов. Все они отражены в модели (рис. 2.15). Важнейшей характеристикой строительного сооружения

является строительная спецификация. Для дальнейшего пространственного моделирования и поиска выделяются размеры и конфигурация сооружения, а также адрес (пространственный индекс).

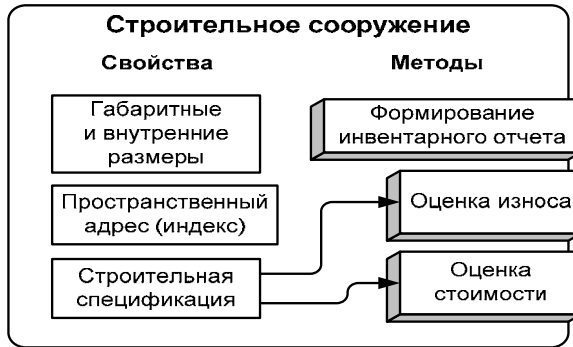


Рис. 2.15. Представление абстрактного сооружения

**Линия** — важнейшая часть инженерной сети — предназначена для передачи продукта на значительное расстояние (рис. 2.16). Как правило, линия состоит из прокладок (труб, хлыстов, отрезков кабелей, отрезков дороги) и мест их стыковки (стыков, муфт, переходов, перекрестков).

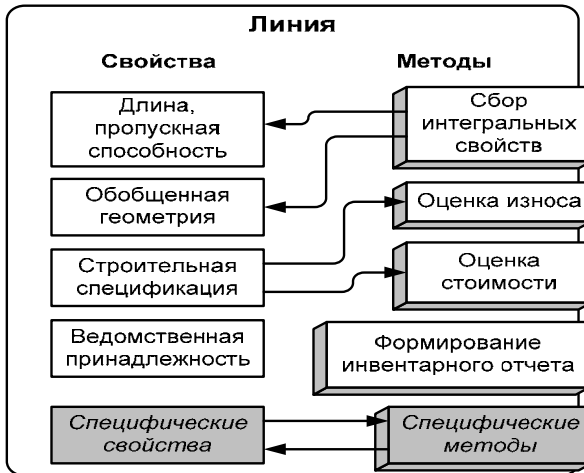


Рис. 2.16. Представление абстрактной линии

Линия характерна тем, что она имеет свойства, получаемые путем интеграции свойств подобъектов, таких как длина, максимальная пропускная способность и т. п. Соответственно, имеет место функция сборки значений данных свойств. Кроме того, линия как протяженный объект имеет геометрическое описание в виде ломаной (осевая линия на плоскости или в пространстве) либо в виде полигона (контур площади, занимаемой линией). Геометрическое описание линии является совокупностью геометрических описаний объектов, входящих в ее состав. Как правило, для удобства их объединяют с некоторым упрощением в обобщенное геометрическое описание всей линии.

**Производственно-технологический участок** (ПТУ) представляет собой совокупность линий, строительных сооружений, отдельных оборудований и ПТУ более мелкого уровня. Он характерен наличием функций управления участком в самом широком смысле. Они используют некоторые параметры управления, такие как нормативные документы, рекомендации, указания диспетчерским службам и т. п. (рис. 2.17).

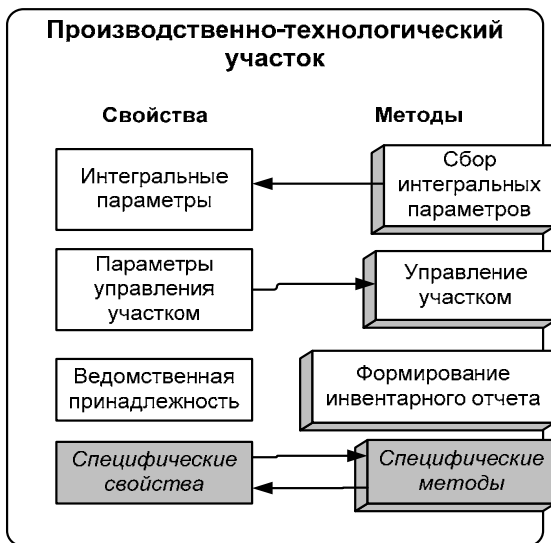


Рис. 2.17. Представление абстрактного производственно-технологического участка

Более детальное моделирование объектов инженерных сетей подразумевает обобщение, объединение однотипных объектов в классы, составление перечня свойств и методов этих объектов, приведение их к единому набору свойств и методов.

### Моделирование жизненного цикла объектов сетей

Моделирование жизненного цикла распадается на три задачи: отслеживание событий; отслеживание параметров; отслеживание переключений состояний сети. Для их решения предлагается использовать соответствующие модели. Основной сущностью для описания жизненного цикла объектов сетей является *событие* — любая совокупность работ (действий), производимых системой управления над объектом сети. Это могут быть измерения параметров, ремонтные работы и другие действия (рис. 2.18).

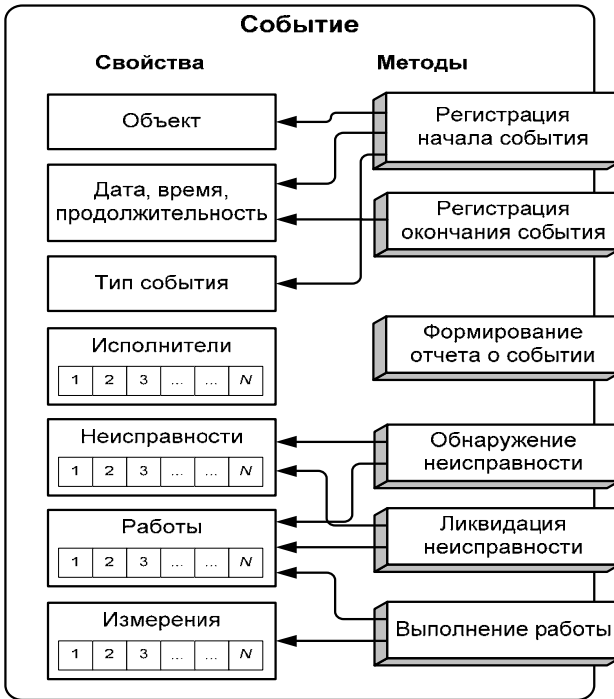


Рис. 2.18. Представление абстрактного события

Любая информация об изменении состояния объекта инженерной сети появляется при возникновении очередного события. Это может быть информация об обнаруженных и устраненных неисправностях, изменении параметров объекта, зафиксированных в результате измерений.

Представление абстрактной работы показано на рис. 2.19.

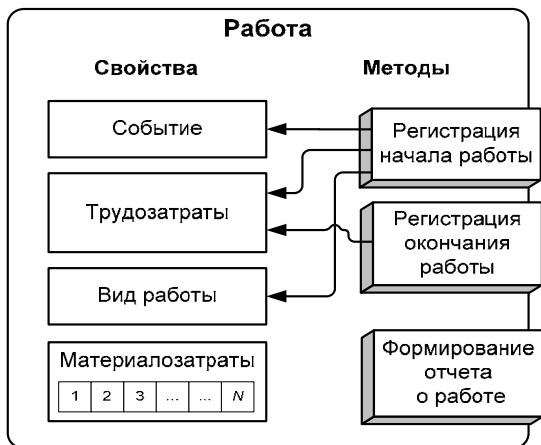


Рис. 2.19. Представление абстрактной работы

Функция «Регистрация окончания работы» может вызвать функции «Обнаружение неисправности» и «Ликвидация неисправности» у события, в составе которого происходит работа. Функции «Регистрация начала работы» и «Регистрация окончания работы» вызываются из функции события «Выполнение работы». Неисправности обнаруживаются в процессе выполнения работ. Некоторые из них могут быть устранены прямо в ходе выполнения работ события, а некоторые могут оставаться неустраненными в течение определенного времени. В соответствии с типом неисправности должна осуществляться автоматическая выдача рекомендаций, например, по составлению графика плановых ремонтов, составлению списка переключений запорного оборудования и т. п. (рис. 2.20).

Измерения значений параметров предлагается моделировать при помощи объекта, изображенного на рис. 2.21. Здесь находит отражение такая информация, как условия проведения измерений, что часто является важным.

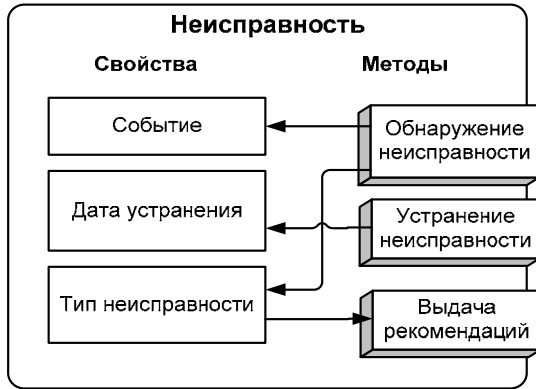


Рис. 2.20. Представление абстрактной неисправности

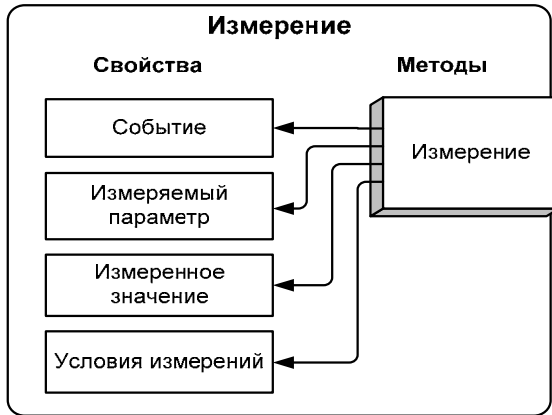


Рис. 2.21. Представление абстрактного измерения параметров

На рис. 2.22 изображена информационная структура, моделирующая жизненный цикл объектов инженерной сети. На графике изображены жизненные линии работ, неисправностей, параметров некоторого объекта инженерной сети. Данная информационная структура позволяет решать две задачи — отслеживание событий и отслеживание параметров.

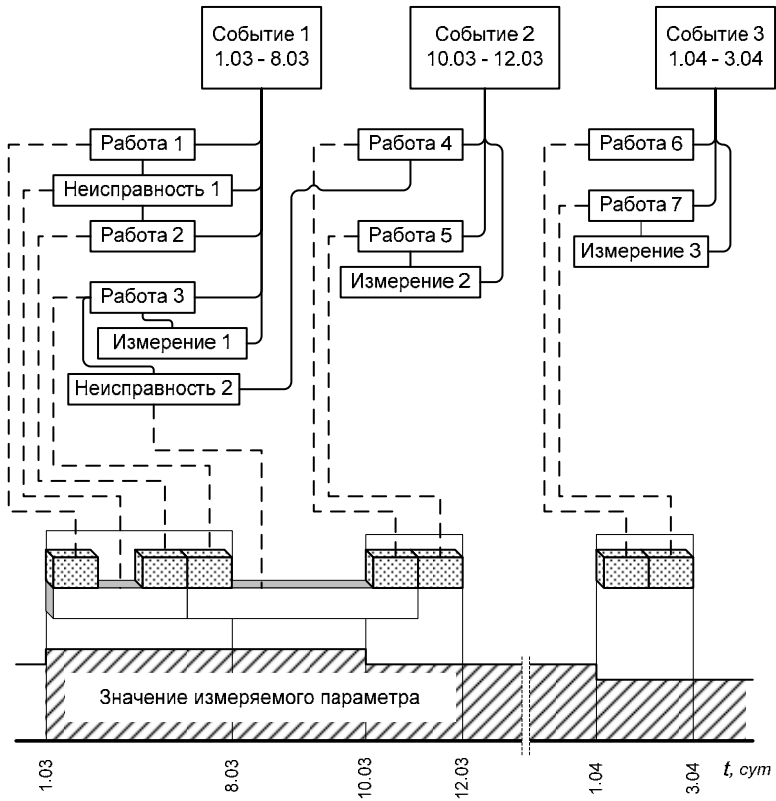


Рис. 2.22. Фрагмент жизненного цикла объекта инженерной сети

Третья задача, имеющая место при моделировании жизненного цикла инженерных сетей, — задача отслеживания переключений состояний объектов. Это могут быть переключения коммутаторов (автоматические или производимые системой управления) либо переключения режимов работы агрегатов инженерной сети. Конечно, их можно рассматривать как события, однако зачастую они происходят автоматически и могут быть весьма частыми. Поэтому предлагается ввести отдельный объект «Изменение состояния», который будет создаваться автоматически при вызове функции «Изменение состояния» у объекта (рис. 2.23).

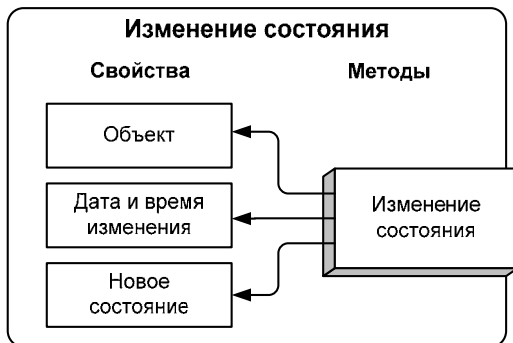


Рис. 2.23. Представление объекта, моделирующего изменение состояния

Все эти модели позволяют полностью описать инженерную сеть как совокупность объектов, выстроенных в некоторую иерархию. Каждый из объектов имеет свою историю, которую можно проследить, совершив соответствующие выборки из множества моделей, описывающих жизненный цикл. Данное описание сети было использовано авторами при построении объектно-ориентированной модели инженерной сети, реализованной на платформе веб-ГИС-сервера, предназначенного для решения задач управления инженерными сетями.

### 2.3. Представления структур данных для решения задач моделирования инженерных сетей

Как уже было сказано выше, для решения задач моделирования инженерных сетей удобно использовать представление сети в виде ориентированного графа. В ходе решения задач моделирования граф может использоваться либо непосредственно как основа для решения, либо как промежуточное представление, по которому строятся системы уравнений или иные математические модели. Использование представления в виде графа позволяет учитывать такое важное свойство сети, как топология (см. гл. 1).

На рис. 2.24 приводится обобщенная иерархия возможных вариантов хранения графа в оперативной памяти для решения задач моделирования инженерных сетей.



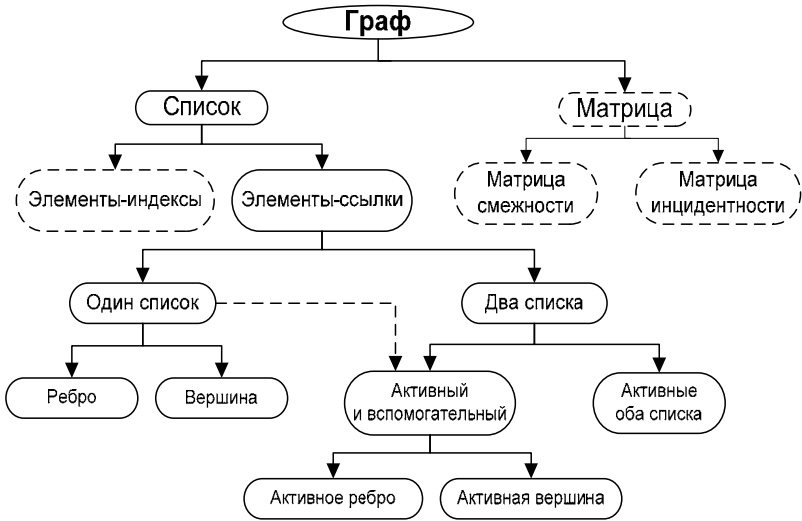


Рис. 2.24. Обобщенная иерархия возможных вариантов хранения графа в оперативной памяти

Хранить данные можно в виде матрицы или списка, это два классических подхода и их преимущества и недостатки хорошо известны.

В матрице можно хранить как информацию о связи вершин (матрица смежности), так и информацию о связи ребер (матрица инцидентности). Хранить граф в виде матрицы удобно, когда количество связей между элементами велико ( $|G|$  сравнимо с  $|U|^2$ , см. параграф 2.1). Если граф неориентированный, то количество хранимых элементов можно уменьшить в два раза, так как матрица получается симметричной относительно главной диагонали. Такой способ хранения возможен при использовании динамических структур данных, но сложнее в реализации. К тому же зачастую приходится иметь дело с ориентированными графами, и, следовательно, использовать всю матрицу целиком.

В случае представления модели дорожной сети типичное количество инцидентных вершине ребер равно четырем (перекресток), поэтому матрица получается сильно разреженной (при количестве ребер, равном 4000, реально использоваться будет меньше 0,1 % занимаемой матрицей памяти). В таких ситуациях целесообразно использовать списки.

Список тоже можно организовать по-разному. Вариации в основном отличаются типом связи между элементами списков. Можно обеспечить связь между элементами списка по их индексам, но тогда пострадает скорость работы. Часть времени будет затрачиваться на поиск элемента по индексу, и, если учитывать количество вызовов этой операции даже при одном проходе по графу, получается ощутимый проигрыш во времени. Гораздо эффективнее использовать связь по ссылкам, когда один элемент списка хранит указатели на другие элементы, связанные с ним, а не индексы элементов. Определить связанность двух вершин с использованием ссылки можно так же быстро, как и при работе с матрицами.

Данные могут храниться в одном или двух связанных списках, в результате чего получаем три различных варианта хранения данных в списке, основанном на ссылках: вершинную структуру, реберную структуру и комбинированную.

### Вершинная структура

Создается список вершин (рис. 2.25) [19]. Каждая вершина помимо общей информации содержит список записей о ребрах, инцидентных этой вершине, которая включает ссылку на соседнюю вершину. Это классический способ хранения, который используется в научных исследованиях при рассмотрении примеров работы с графами. В этом способе хранения имеется недостаток — дублирование информации о ребрах (она будет храниться для обеих вершин ребра).

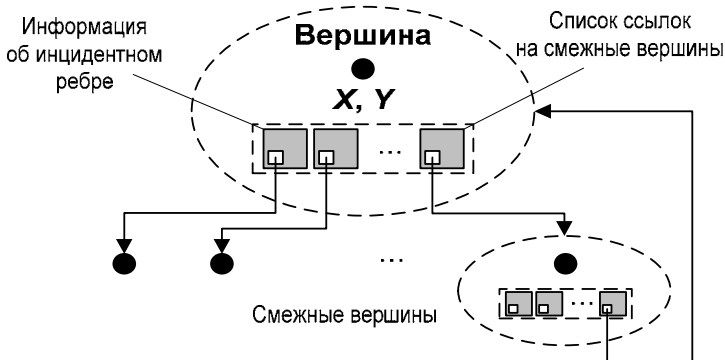


Рис. 2.25. Структура хранения графа в виде списка вершин

Отмеченный недостаток устраняется путем введения второго вспомогательного списка (списка ребер). При этом размер занимаемой списком памяти уменьшится без вреда быстродействию. Таким образом, вершина теперь будет хранить список записей, состоящий из двух ссылок (на соседнюю вершину и на данные о ребре, лежащем между ними). Такой способ называется вершинной структурой со вспомогательным списком ребер.

### Реберная структура

При этом способе хранения за основу принимается ребро. Каждый элемент списка хранит данные и ссылки на инцидентные ребра (рис. 2.26).

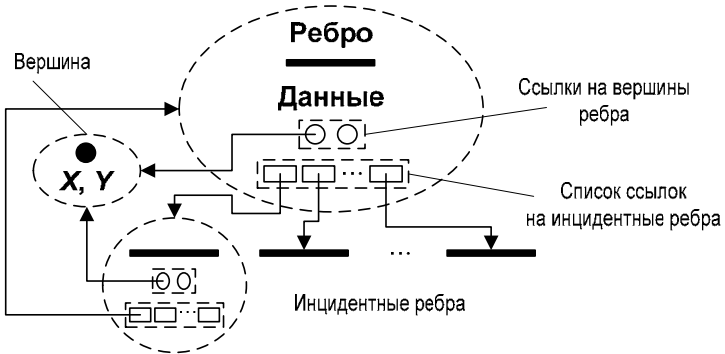


Рис. 2.26. Структура хранения графа в виде списка ребер

В базовом варианте представления информация о вершинах может отсутствовать совсем. Если же она необходима, то ее можно хранить в списке вместе с указателями на ребра. Такое хранение вершин приводит к еще большему дублированию данных, чем в базовой вершинной структуре, так как одна вершина связана обычно с несколькими ребрами. Чтобы избежать этого, можно, как и в предыдущем случае, завести вспомогательный список вершин, а в элементах списка ребер хранить указатели на эти вершины (см. рис. 2.26) [19].

При этом получаем относительную независимость ребер от вершин, т. е. вершины используются в качестве дополнительных данных и от них можно избавиться, не разрывая связь элементов списка ребер. В этом и заключается основное достоинство способа. Недостаток же состоит в том,

что такой список относительно трудно заполнять (если добавляется ребро, то нужно сразу добавить и остальные ребра, чтобы сослаться на них). К тому же возникают некоторые трудности с выделением ребер, инцидентных конкретной вершине. Также необходимо признать, что чисто реберная структура (чем бы она ни дополнялась) больше предназначена для хранения неориентированных графов и применима в специфических задачах.

**Комбинированная структура** содержит одновременно два равноценных списка — список вершин и список ребер (рис. 2.27). Причем оба списка являются «активными», т. е. содержат ссылки на элементы другого.

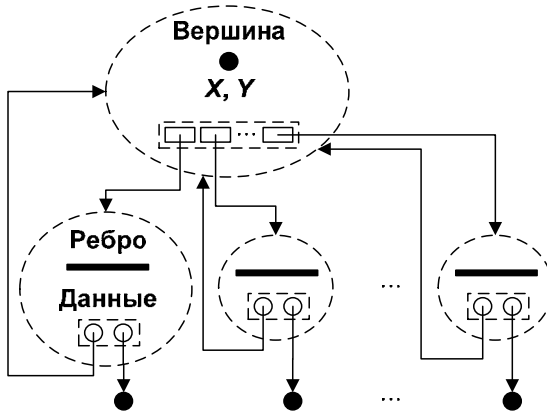


Рис. 2.27. Структура хранения графа в виде комбинированного списка

Каждая вершина помимо общей информации содержит список ссылок на инцидентные ей ребра, а ребро содержит ссылки на две вершины, которым оно инцидентно (см. рис. 2.27) [19]. Такое хранение данных является наиболее универсальным вариантом, так как позволяет работать как с ребрами, так и с вершинами. Добавление элементов в список и получение доступа к одним элементам из других происходит без особых проблем. При таком способе хранения каждый элемент содержит лишь относящиеся к нему данные, что наиболее близко соответствует принципу объектно-ориентированного подхода и является наи-

более правильным вариантом (по мнению автора), при этом структура не занимает больше места в памяти, чем при хранении предыдущими способами. Единственный недостаток — это жесткая взаимосвязь элементов списков вершин и ребер, но именно такая организация и дает вышеперечисленные преимущества. На рис. 2.28 представлен пример хранения графа (рис. 2.28, *a*) в памяти с помощью реберного списка (рис. 2.28, *б*) и комбинированного списка (рис. 2.28, *в*).

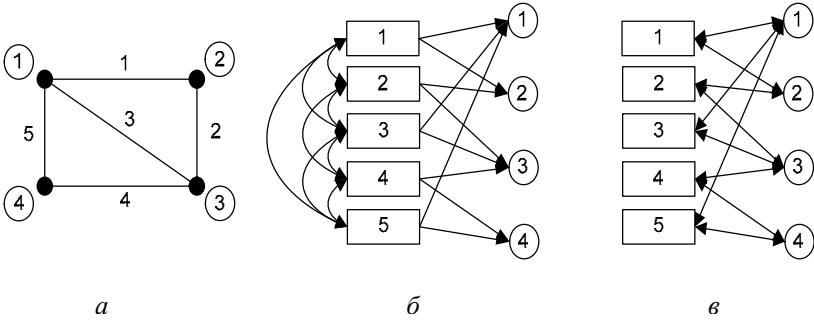


Рис. 2.28. Сравнение структур хранения графа: *a* — пример графа; *б* — реберный список; *в* — комбинированный список

Анализ трех способов представления графа позволяет сделать вывод о том, что вершинная и реберная структуры применимы лишь для узкоспециализированных задач (например, при необходимости хранения информации только о вершинах или только о ребрах), а комбинированная структура является универсальным способом и рекомендуется к применению всегда при использовании информации и о вершинах, и о ребрах [19].

Так как количество ссылок и занимаемой памяти одинаково во всех трех способах, то следует выбирать структуру по другим критериям, таким как удобство построения графа, удобство работы со структурой данных и специфика задачи. В рассматриваемом классе задач при работе с ребрами графа лучше всего подходят второй и третий способы. По общей идеологии задачи можно было бы отбросить вершины как таковые и работать только с ребрами, но необходимо все же знать координаты вершин на карте, поэтому здесь нагляднее было бы использовать реберную структуру с дополнительным списком вершин. С другой стороны, в разнообразных алгоритмах на графах (поиск кратчайшего пути и др.)

требуется работать как с вершинами, так и с рёбрами (почти в равной степени), поэтому все-таки лучше использовать комбинированную структуру. В пользу этого свидетельствует также простота построения списков и гибкие возможности доступа к их элементам. Практически не возникает никаких затруднений в получении информации об инцидентных ребрах и смежных вершинах для какой-либо вершины и аналогичных данных об окружении ребра.

Обобщая вышеприведенные рассуждения, можно заключить, что комбинированная структура является эффективным представлением данных для моделирования процессов транспортировки целевого продукта в инженерных сетях (сетях водоснабжения и водоотведения, сетях теплоснабжения, газопроводах и нефтепроводах), а также для различных режимов расчета электрических сетей.

### Упрощение графовой структуры

Представляется целесообразным производить предварительное упрощение графа с целью уменьшения его размерности (и, как следствие, уменьшения размерности задачи потокораспределения). Первый вид упрощения состоит в свертке цепочки ребер, при которой цепочки однотипных ребер заменяются на одно ребро (рис. 2.29) [4].

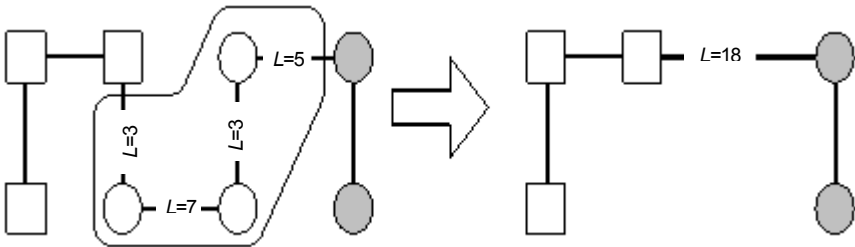


Рис. 2.29. Упрощение структуры графа (свертка цепочки ребер)

Кроме свертки цепочки ребер для уменьшения размерности графа предлагается разбивать задачу по построению моделей инженерных сетей на ряд задач построения моделей инженерных сетей меньшей размерности (рис. 2.30) [4].

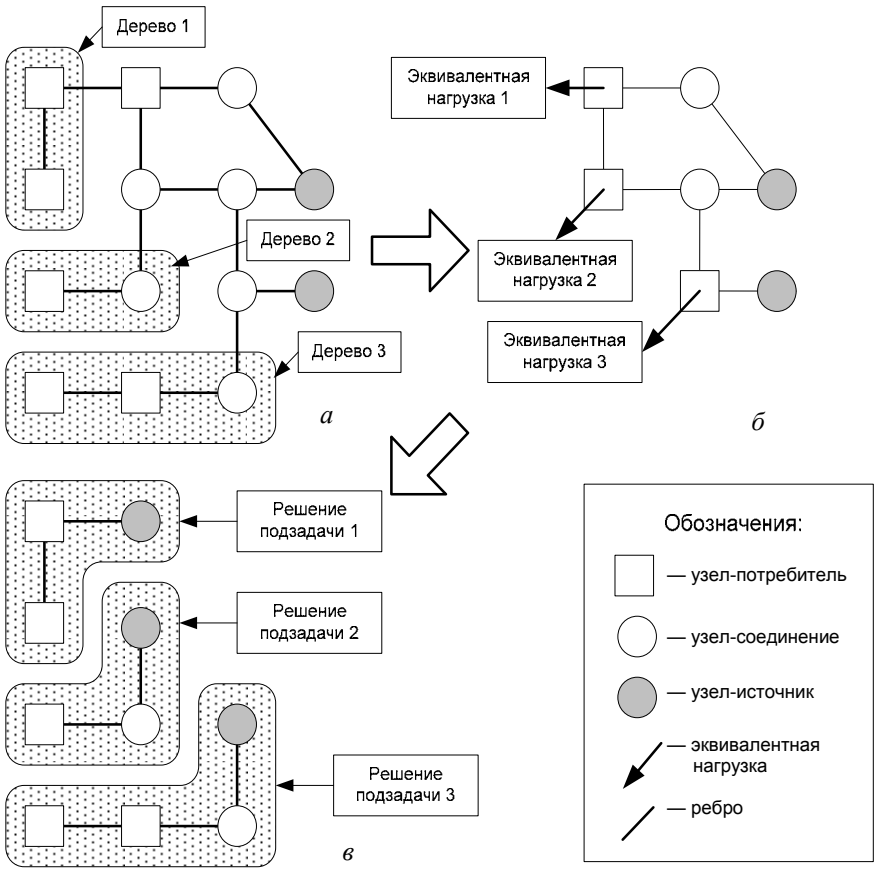


Рис. 2.30. Решение задач моделирования при эквивалентировании односвязных компонентов графа сети:  
*a* — выделение деревьев; *б* — эквивалентирование;  
*в* — моделирование на «подсетях»

Для построения модели сети меньшей размерности в графе выделяются подграфы, представляющие собой деревья, или односвязные компоненты, не содержащие источников (см. рис. 2.30, *a*). Каждое такое дерево имеет корневую вершину, через которую осуществляется связь

с остальным графом. Дерево удаляется из исходного графа, а его корневой вершине ставится в соответствие суммарная потребность потребителей удаленного дерева (эквивалентная нагрузка) (см. рис. 2.30, б). После удаления всех односвязных компонентов из графа решаются задачи моделирования инженерной сети. Далее задачи моделирования решаются отдельно для всех деревьев, исходными значениями для которых будут значения, полученные на предыдущем этапе обобщенного моделирования (см. рис. 2.30, в) [4].

Данные подходы можно применять как в алгоритмах, непосредственно базирующихся на графах, так и в алгоритмах, использующих граф как промежуточное представление. Для выделения односвязных компонентов можно воспользоваться простым алгоритмом, основанным на нумерации вершин графа методом поиска в ширину.

Как правило, такое упрощение графа инженерной сети дает снижение размерности задачи в несколько раз для разветвленных сетей. Однако в настоящее время большое количество сетей делается закольцованными с целью увеличения надежности. В таком случае даже после проведения эквивалентирования деревьев граф сети остается достаточно сложным.



# ТЕХНОЛОГИИ ФУНКЦИОНИРОВАНИЯ ИНЖЕНЕРНЫХ ГИС

---

### 3.1. История использования ГИС

Эффективное ведение процесса управления инженерными сетями обеспечивается при использовании современных информационных технологий, предоставляющих различные варианты доступа к информации об инженерной инфраструктуре предприятия:

- 1) использовании локальной ГИС;
- 2) использовании многопользовательской ГИС;
- 3) использовании многопользовательской ГИС и распределенного веб-доступа.

**Использование локальной ГИС.** На предприятиях, занимающихся эксплуатацией инженерных сетей, все еще можно встретить самую архаичную схему доступа, при которой пользователи получают фрагменты плана местности с инженерными сетями только на бумажном носителе (рис. 3.1) [20].

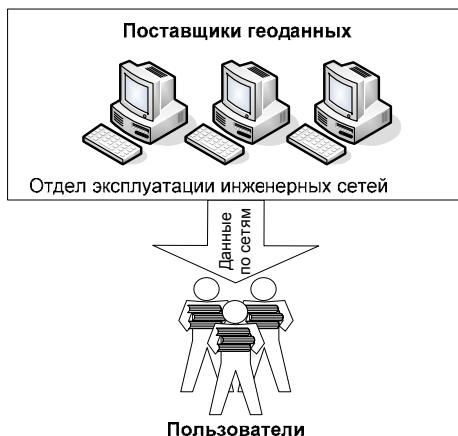


Рис. 3.1. Классическая схема доступа к данным по инженерным сетям

На предприятии, только вступившем в процесс автоматизации производства, ведение электронного представления планов инженерных сетей может осуществляться с использованием локальных инструментальных ГИС, которые позволяют хранить пространственные данные либо в виде обычных файлов, либо в локальных СУБД (рис. 3.2) [20].

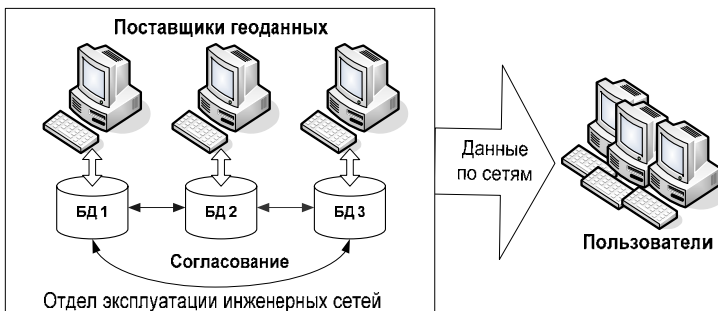


Рис. 3.2. Разработка и получение данных электронного представления инженерных сетей при использовании локальной ГИС

Этот подход имеет ряд существенных недостатков:

- **неполнота атрибутивных описаний объектов инженерных сетей.** Большинство инструментальных ГИС не предназначены для поддержки полного атрибутивного описания пространственных объектов инженерных сетей. Как правило, объект может содержать несколько простых свойств. Однако в действительности объекты инженерной инфраструктуры имеют более обширные атрибутивные описания;

- **сложность синхронизации версий планов по инженерным сетям.** Из-за отсутствия единого хранилища, а также пространственных и атрибутивных данных возникает проблема синхронизации изменений в планах по инженерным сетям. На каждом рабочем месте находится копия набора данных. После редактирования необходимо копировать измененные данные на все рабочие места, а также согласовывать изменения, производимые каждым пользователем.

Необходимость решения данных проблем ведет к поиску иных методов решения задачи ведения электронных планов по инженерным сетям. Наиболее приемлемым становится применение технологии «клиент-сервер».

**Использование многопользовательской ГИС.** Вместо локальной системы, функционирующей на одном рабочем месте, используется централизованная многопользовательская система, в которой множество пользователей могут одновременно работать в едином информационном пространстве вычислительной сети (рис 3.3) [20].

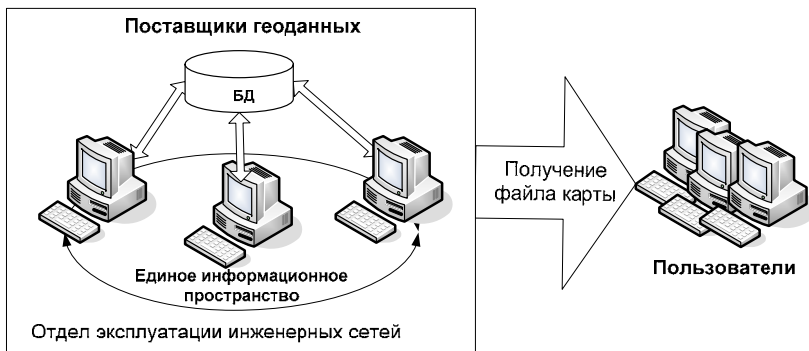


Рис. 3.3. Разработка и получение данных электронного представления инженерных сетей при использовании многопользовательской ГИС

Многопользовательские ГИС позволяют работать с данными нескольких серверов одновременно, причем в одной карте можно объединять данные из разных источников и различных серверов, локальных пространственных баз, а также из файлов на локальном диске. Данный метод повышает эффективность процесса ведения электронных планов, однако остаются нерешенными следующие проблемы:

- **отсутствие оперативности.** Пользователи не имеют оперативного доступа к данным по инженерным сетям. Это затрудняет координацию работ инженерных подразделений предприятия;
- **отсутствие ограничения доступа к данным.** Пространственные и атрибутивные данные доступны всем пользователям для чтения и редактирования. Необходимы временные затраты на предоставление ограниченного варианта электронного плана инженерных сетей;
- **отсутствие у пользователей необходимых навыков работы с ГИС.** Для выполнения простых операций анализа пространственных данных (измерения расстояния, площади, периметра) необходимо иметь опыт работы с ГИС, уметь пользоваться стандартным инструментарием.

**Использование многопользовательской ГИС и распределенного веб-доступа.** Применение технологии публикации пространственных данных обеспечивает централизованное хранение, анализ и предоставление пространственных данных в корпоративной сети и в сети Интернет для удаленных пользователей (рис. 3.4) [20].

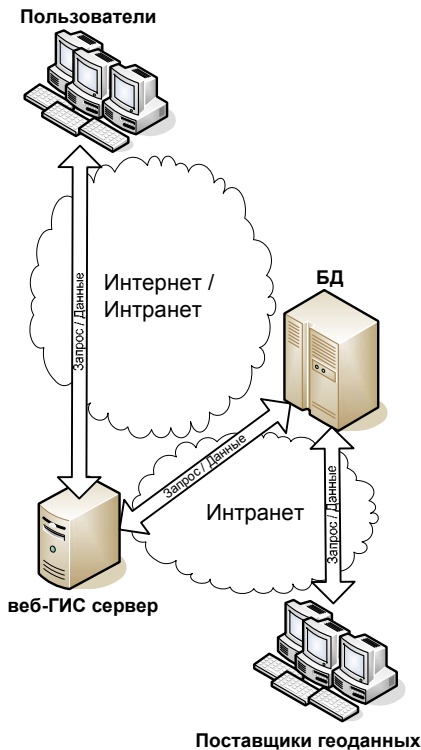


Рис. 3.4. Разработка и получение данных электронного представления инженерных сетей при использовании интернет-ГИС

При данном подходе инженерные службы и подразделения могут пользоваться электронными планами инженерных сетей из любой точки предприятия, где имеется доступ к корпоративной вычислительной сети.

Поставщики геоданных взаимодействуют с единым хранилищем пространственных и атрибутивных данных (БД). За публикацию пространственных данных отвечает веб-ГИС-сервер, который является связующим звеном между пользователями и электронным генеральным планом.

Пользователям, имеющим стандартный браузер (тонкий клиент), не требуется устанавливать какое-либо геоинформационное программное обеспечение на свой компьютер, так как вся работа с электронными планами по инженерным сетям осуществляется в окне браузера. Карта может передаваться как в векторном, так и в растровом виде.

В случае использования веб-ГИС-сервера процесс получения электронных данных значительно упрощается.

## **3.2. Информационные технологии построения веб-ориентированной ГИС**

Идея публикации геоинформационных данных в сети Интернет возникла в 90-х годах XX в. Фактически датой рождения веб-картографии можно считать 1993-й год, когда впервые был запущен веб-сервис Хехох PARC Map Viewer [21]. Но реально такая возможность появилась сравнительно недавно. Переломным стал 2005-й год, когда компания Google практически одновременно запустила два глобальных картографических сервиса — Google-Maps и Google-Earth [21]. Это стало возможным благодаря появлению новых высокопропускных каналов связи и развитию микроэлектронной базы ЭВМ.

Первоначально функции веб-ГИС сводились лишь к просмотру фиксированных изображений, представляющих карты в форматах GIF, JPEG [22]. Интерфейс взаимодействия пользователя с веб-сервером был ограничен и сводился к простому выбору требуемого изображения. Преимуществами такого метода публикации данных являются: простота публикации, низкие требования к серверу, канал связи низкой пропускной способности, наличие на клиентском месте только веб-браузера.

Затем появляются системы просмотра картографической информации, использующие механизм тематических запросов. На сервере сети, где функционирует веб-сервер, организуется база данных, представляющая собой набор тематических категорий. Каждая категория содержит

определенный набор тематических карт, хранящихся в растровых форматах GIF, JPEG. Пользователь, попадая на такой сервер, должен выбрать по базе данных тему и регион, охватываемый картой, а также набор дополнительных условий. Результатом запроса к БД является отображение того или иного изображения карты на экране компьютера пользователя. По сравнению с предыдущим методом данный вариант предъявляет более высокие требования к серверу, но обеспечивает более структурированный подход к отображению картографических данных [22].

Дальнейшее развитие геоинформационных систем в Интернете сопровождается созданием интерактивных сред взаимодействия клиентских приложений с геоинформационным сервером [23], суть которых состоит в предоставлении возможности пользователю самостоятельно выбирать на карте участки для отображения на своем компьютере. На сервере размещается набор серверных программ, обеспечивающих взаимодействие с клиентом, анализ действий клиента и создание растровой картинки на область, указанную пользователем. Модель взаимодействия клиента с сервером представляет собой следующее: пользователь видит генерализованный участок карты и, выбирая более мелкие участки, получает все более детальное отображение местности. Такое интерактивное взаимодействие требует более мощных серверов.

Параллельно с растровой технологией развивается *технология передачи векторных данных*, упрощающая работу сервера (не требуется преобразование векторных данных в растровый формат) и при малой насыщенности карты объектами уменьшающая объем передаваемых данных. Для отображения данных, полученных от сервера, обычно используется клиентское программное обеспечение — программы просмотра (viewers), которые реализуются как ActiveX-компонент, Plug-in или Java-апплет и способны встраиваться в прикладные программы или веб-браузер.

Использование специальных компонентов переносит часть функций по обработке данных на сторону клиента, обеспечивая пользователя расширенными функциями (манипулирование тематическими слоями карты, пространственный анализ, получение информации об объекте и пр.). Вместе с тем выявляются и недостатки данного подхода. Программа просмотра требует для первоначальной установки и настройки ее загрузки с сервера. В дополнение к этому накладываются ограничения на тип используемого веб-браузера. Это осложняет быстрый доступ к пространственным данным с любого компьютера, подключенного к сети. При боль-

шой насыщенности области карты может передаваться гораздо больший объем информации, чем при использовании растрового варианта. Такой режим работы удобен только при наличии скоростного канала связи и достаточно мощных компьютеров на клиентских местах. Поэтому векторный вариант больше подходит для использования в корпоративных сетях.

При последующем развитии технологий появилась потребность просмотра как географической (графической) информации, так и привязанной к ней атрибутивной информации. Если в векторной технологии этот подход было реализовать довольно просто, то при использовании растра неизбежно использование на стороне клиента дополнительных средств (скриптов) для отправки запроса на сервер. При этом сервер должен обеспечивать не только высокую скорость растеризации/преобразования исходной графической информации, но и высокоскоростную обработку атрибутивной информации, хранящейся в базе данных.

Использование векторного подхода позволяет получить из программы просмотра идентификатор выбранного объекта и затем запросить атрибутивную информацию с сервера любым из возможных способов. В случае указания объекта на растровом изображении карты существует только лишь один подход — передача ГИС-серверу координат объекта, по которому запрашивается информация (в виде HTTP<sup>1</sup>-запроса). В этом случае сервер своими средствами выделяет нужный объект, учитывая доступность различных слоев карты и прочие ограничения, и возвращает ответ клиенту по протоколу HTTP. Формат взаимодействия клиента и сервера в сетевой ГИС определяется стандартами организации Open GIS Consortium, Inc (<http://www.opengis.org/>), которых придерживается большинство разработчиков, представляющих свои продукты на рынке сетевых ГИС-технологий. Ответ возвращается в формате GML<sup>2</sup> и подлежит дополнительной обработке или требует задания схемы интерпретации ответа браузером.

---

<sup>1</sup> HTTP — HyperText Transfer Protocol (протокол передачи гипертекста). Основной HTTP является технология «клиент-сервер», предполагающая существование потребителей (клиентов), инициирующих соединение и посылающих запрос, и поставщиков (серверов), ожидающих соединения для получения запроса, которые производят необходимые действия и возвращают обратно сообщение с результатом.

<sup>2</sup> GML (Geography Markup Language) — язык географической разметки, разрабатываемый Open GIS Consortium.

Примером взаимодействия клиента с таким ГИС-сервером может быть следующая схема. Пользователю предлагается обобщенная (генерализованная) карта, по которой он может выбрать участок для последующей детализации. Также на странице сервера предлагается набор функций для поиска информации по атрибутивному признаку. Это может быть как список номеров колодцев, инженерных сооружений, так и полей для ввода критериев поиска.

После набора необходимых данных для атрибутивного поиска пользователю предоставляется либо искомый объект в графической форме, либо список найденных объектов. Выбирая объекты, пользователь получает их картографическое изображение и имеет возможность просмотра атрибутивной информации, присоединенной к графическому объекту. Следует обратить внимание, что при использовании интернет-технологий появляется возможность интеграции нескольких различных проектов с ГИС-модулем, что значительно расширяет базовые возможности геоинформационной системы.

Говоря о разработке интернет-ГИС, нельзя не затронуть вопрос о средствах для обеспечения функциональности клиентских приложений. Используемые ActiveX<sup>1</sup>-компоненты уже содержат в себе минимальную функциональность и при этом предоставляют набор интерфейсных функций (API<sup>2</sup>) для настройки и управления. Данные функции могут быть использованы в таких приложениях, как JavaScript<sup>3</sup>, JScript<sup>4</sup>, а также в

---

<sup>1</sup> ActiveX — средство, при помощи которого Internet Explorer (IE) использует другие приложения внутри себя. Элементы управления ActiveX активизируются при щелчке по такому объекту на веб-странице.

<sup>2</sup> API (Application programming interface) — набор готовых классов, функций, структур и констант, предоставляемых приложением (библиотекой, сервисом) для использования во внешних программных продуктах.

<sup>3</sup> JavaScript — объектно-ориентированный скриптовый язык программирования. JavaScript обычно используется как встраиваемый язык для программного доступа к объектам приложений. Наиболее широкое применение находит в браузерах как язык сценариев для придания интерактивности веб-страницам.

<sup>4</sup> JScript — скриптовый язык программирования компании Microsoft. JScript во многом аналогичен языку JavaScript компании Netscape, однако, помимо добавления клиентских скриптов на веб-страницы, JScript может использоваться и для других целей, например автоматизации администрирования систем Microsoft Windows, создания страниц ASP.



VBScript<sup>1</sup>-приложения. Задачи, выполняющие сложные расчеты и оперирующие большими объемами данных (например, задачи на инженерных сетях), требуют использования серверных приложений.

Наиболее популярным подходом является написание CGI-<sup>2</sup>, ISAPI/NSAPI-приложений<sup>3</sup> или применение технологий PHP<sup>4</sup>, ASP<sup>5</sup> и Perl<sup>6</sup>. Perl<sup>6</sup>. Разработка первых возможна с использованием большинства современных высокоуровневых средств разработки (Borland Delphi, Microsoft Visual C++) и предоставляет широкие возможности в использовании. Последние обеспечивают простоту взаимодействия частей проекта и более быструю разработку приложений, но обладают меньшей гибкостью при реализации алгоритмов и поэтому не подходят для ряда задач.

Анализ архитектуры систем ведущих производителей позволяет сделать вывод о том, что все веб-проекты строятся примерно по одной схеме с применением фирменных технологий. Концепция строится на классической трехзвенной клиент-серверной архитектуре. Распределенная информационная система выполняет следующие функции:

---

<sup>1</sup> VBScript — скриптовый язык программирования, интерпретируемый компонентом Windows Script Host. Он широко используется при создании скриптов в операционных системах семейства Microsoft Windows. VBScript был создан компанией Microsoft как замена устаревшему пакетному языку, интерпретируемому приложением command.com.

<sup>2</sup> CGI (Common Gateway Interface) — стандарт интерфейса, используемый для связи внешней программы с веб-сервером. Программу, которая работает по такому интерфейсу совместно с веб-сервером, принято называть шлюзом, хотя многие предпочитают названия «скрипт» (сценарий) или «CGI-программа».

<sup>3</sup> ISAPI — стандарт Internet Server API, изначально созданный как Microsoft Information Server API, но в дальнейшем предложенный в качестве открытого стандарта. NSAPI — стандарт Netscape Server API, используемый для взаимодействия с серверами компании Netscape.

<sup>4</sup> PHP (Personal Home Page Tools) — скриптовый язык программирования общего назначения, интенсивно применяемый для разработки веб-приложений. В настоящее время поддерживается подавляющим большинством хостинг-провайдеров и является одним из лидеров среди языков программирования, применяющихся для создания динамических веб-сайтов.

<sup>5</sup> ASP (Active Server Pages) — первая технология компании Microsoft, позволяющая динамически создавать веб-страницы на стороне сервера.

<sup>6</sup> Perl — высокоуровневый интерпретируемый динамический язык программирования общего назначения.

- 1) диалог с пользователем (Presentation Logic);
- 2) прикладные функции (Business Logic);
- 3) обработку данных внутри приложения (Database Logic);
- 4) управление информационными ресурсами (Database Management System);
- 5) служебные функции (связующее звено).

Основу трехуровневой (трехзвенной) модели составляет классическая модель «клиент-сервер». В трехуровневой модели вводится дополнительное звено между клиентом и сервером — сервер приложений, который принимает на себя функциональную нагрузку бизнес-логики системы. Сервер баз данных (БД) при этом разгружается и исполняет исключительно функции управления данными. Помимо этого, введение третьего звена делает систему более гибкой и масштабируемой.

В зависимости от распределения функциональной нагрузки различают две модели передачи данных: тонкий клиент и толстый клиент (рис. 3.5).

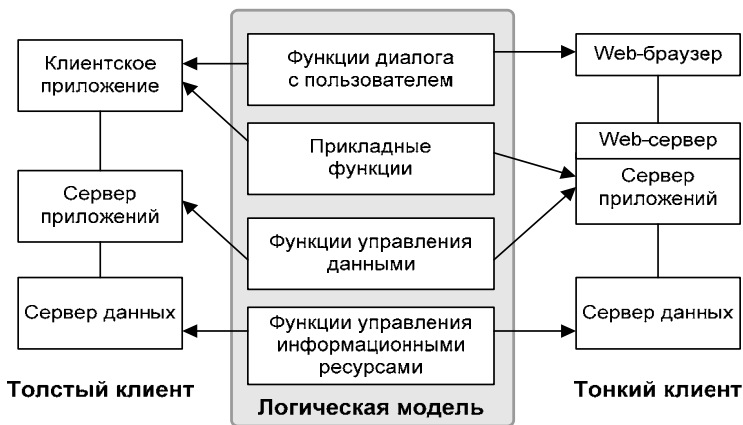


Рис. 3.5. Модели сетевой ГИС

Классический анализ этих моделей приводится во многих работах, в рамках данного исследования рассматривается их использование применительно к передаче пространственных данных. Последовательность

выполнения операций и задействованные при этом блоки системы согласно стандарту OGC представлены в виде схемы на рис. 3.6.

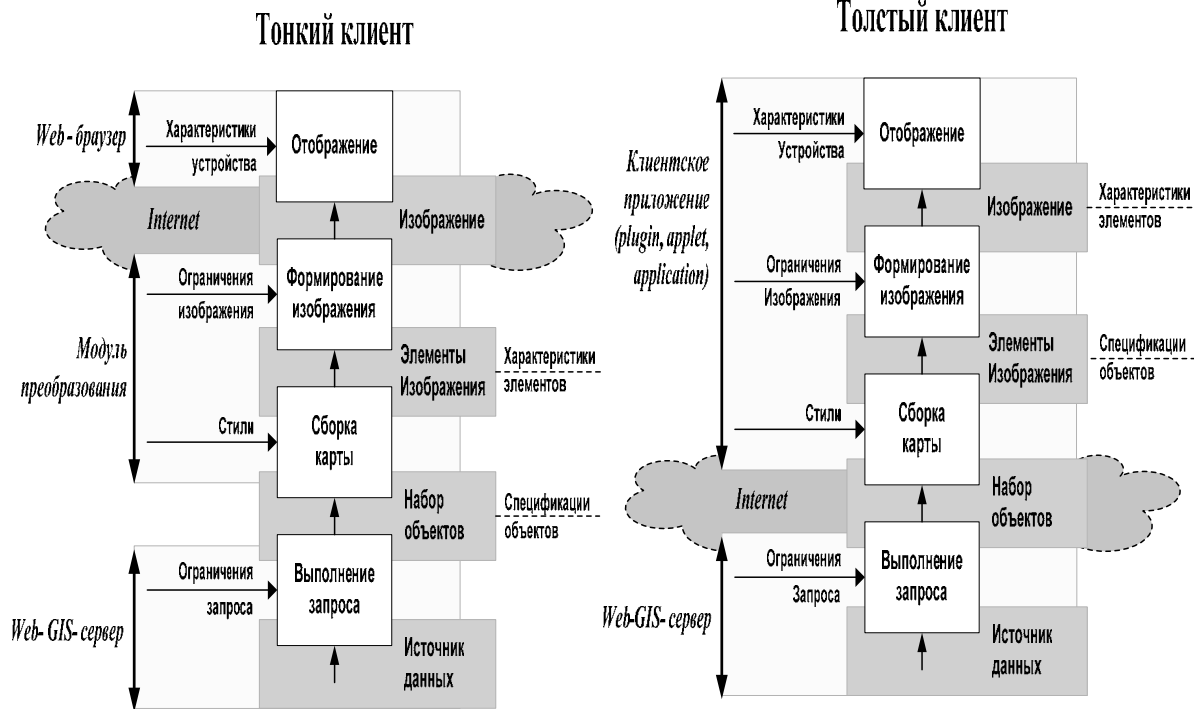


Рис. 3.6. Схема распределения функций системы при передаче пространственных данных через Интернет

На представленном рисунке 3.6 показана часть процесса передачи данных (ответ сервера на запрос пространственных данных), на которой четко разделены функции клиента и сервера.

В системе с трехзвенной архитектурой можно выделить несколько основных частей, представленных на обобщенной схеме взаимодействия компонентов сетевой ГИС (рис. 3.7).

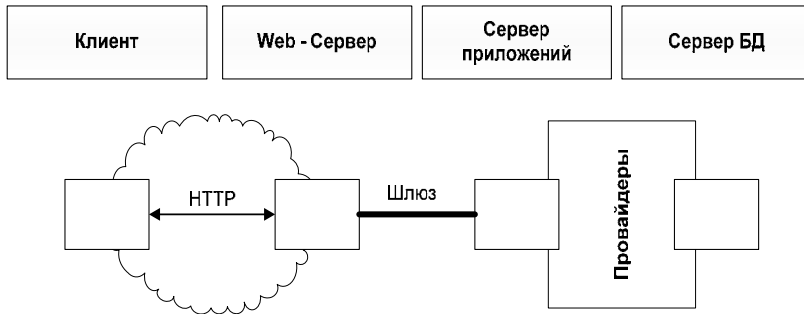


Рис. 3.7. Обобщенная схема взаимодействия компонентов сетевой ГИС

Детальная схема взаимодействия компонентов сетевой ГИС показана на рис. 3.8.

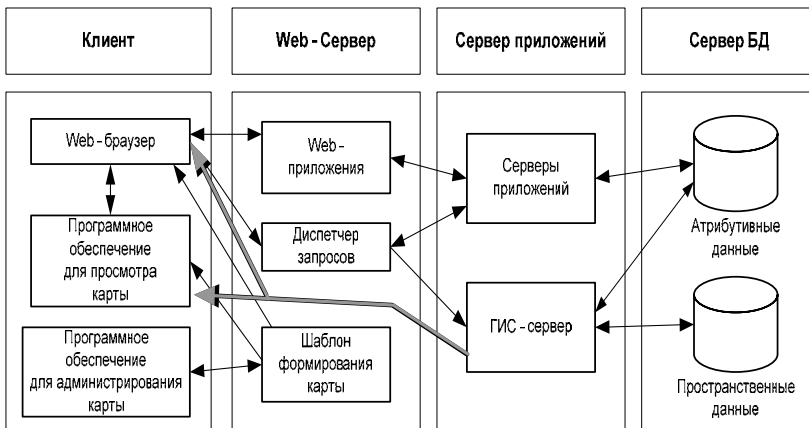


Рис. 3.8. Схема взаимодействия компонентов сетевой ГИС

**Клиентское приложение** предназначено для просмотра карты и представлено веб-браузером (тонкий клиент), модулем отображения данных, встраиваемым в веб-браузер, или пользовательским приложением (толстый клиент). Не стоит оставлять в стороне и средства для редактирования, администрирования и сборки карты, которые также работают с сервером удаленно, а следовательно, относятся к клиентским приложениям. Любое клиентское приложение (кроме растрового варианта) получает от сервера данные по определенному указываемому в запросе шаблону [23]. Файлы шаблона хранятся на сервере и включают информацию о настройках слоев карты, необходимых для ее корректного отображения. Эта информация включает имена слоев, цвета и типы линий, путь к источнику данных и другую информацию о карте.

**Веб-сервер** выполняет функции посредника между клиентскими и серверными приложениями. Используется для передачи запросов и данных по протоколу HTTP. Любой современный веб-сервер, использующийся для задач сетевой ГИС, позволяет выполнять создаваемые специально приложения, которые служат для формирования и обработки запросов, работы с дополнительными атрибутивными данными, решения прикладных задач, расширения функциональности ГИС и др.

В случае сетевой ГИС на уровне веб-сервера располагается также диспетчер запросов. Этот функциональный модуль служит для поддержания связи клиента с ГИС-серверами. Он принимает запросы на картографические данные, приходящие на веб-сервер от различных клиентских приложений, затем их выстраивает, распределяет и передает для обработки на ГИС-сервер. Многоуровневая архитектура и наличие такого звена, как «диспетчер» позволяют создать систему с несколькими распределенными ГИС-серверами (рис. 3.9).

**Сервер приложений** в концепции сетевой ГИС представляет собой веб-GIS-сервер. Данный компонент состоит из нескольких взаимодействующих частей. В технологиях, предлагаемых разными фирмами, его структура различна, но имеется несколько общих блоков. Основным блоком является «Исполнительная служба». Этот блок получает и обрабатывает запросы от диспетчера, оперируя пространственными и атрибутивными данными, полученными из некоторых источников данных. Затем данные форматируются и посылаются через веб-сервер запросившему их клиенту по протоколу HTTP. При этом есть различия при передаче растровых и векторных данных от сервера на сторону клиента.

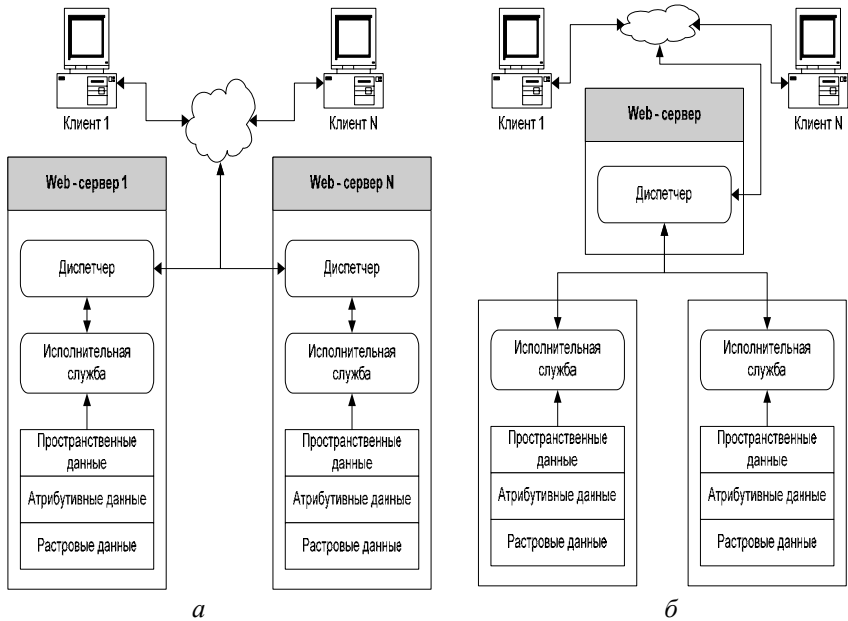


Рис. 3.9. Распределенное серверное решение: *a* — получение клиентом данных с разных серверов; *b* — распределение диспетчером запроса на разные исполнительные службы

При передаче растровых данных атрибутивная информация по объектам карты остается недоступной для пользователя и ее необходимо либо загружать, используя дополнительные средства, либо запрашивать отдельно в процессе работы с картой. Для этого в состав веб-GIS-сервера включается дополнительный модуль по обработке картографических данных для растрового варианта карты «Служба растрового преобразования». Этот модуль, как правило, реализуется в виде приложения какого-либо сервера приложений JRun (Sun), Apache TomCat (свободно распространяемое ПО) и служит для формирования растрового варианта карты и возвращения атрибутивной информации. При поступлении запроса от диспетчера это приложение загружает с веб-сервера файл сборки (шаблон карты), а затем производит запрос данных по тому же механизму, что и клиентские приложения (т. е. посредством HTTP-запроса).

После получения данных модулем они обрабатываются и извлеченная атрибутивная информация в формате GML передается в запросившее ее приложение. Аналогичная схема используется и при работе с мобильными клиентскими приложениями, где требуется адаптация данных под конкретное используемое устройство. Для получения данных обычно используется механизм ODBC<sup>1</sup> или OLEDB<sup>2</sup>.

**Сервер баз данных.** Этот компонент вносит еще большую вариативность в технологию, представленную на рис. 3.5, так как «Исполнительная служба» может запрашивать данные с удаленных серверов БД, используя механизмы поставщиков данных. Модуль обработки пространственных данных может быть частью ГИС-сервера. Тогда он ответственен за загрузку всей пространственной информации, в том числе и хранящейся в файловом виде. Но наиболее правильным является пространственный компонент как часть системы управления БД (СУБД).

### 3.3. Технология хеширования данных

На современном этапе развития геоинформационных систем широко распространяется подход к хранению пространственных данных совместно с атрибутивными в клиент-серверных СУБД. Лидером на рынке программного обеспечения, предназначенного для хранения пространственных данных, можно смело назвать СУБД Oracle с ее опцией Oracle Spatial<sup>3</sup>. На этапе подготовки электронных чертежей присутствует

---

<sup>1</sup> ODBC (Open Database Connectivity) — программный интерфейс (API) доступа к БД, разработанный фирмой Microsoft в сотрудничестве с Simba Technologies.

<sup>2</sup> OLEDB (Object Linking and Embedding, Database) — набор интерфейсов, основанных на COM (Component Object Model — технологический стандарт от компании Microsoft, предназначенный для создания программного обеспечения на основе взаимодействующих распределенных компонентов, каждый из которых может использоваться во многих программах одновременно и позволять приложениям обращаться к данным, хранимым в разных источниках информации или хранилищах данных с помощью унифицированного доступа).

<sup>3</sup> Oracle Spatial — опция Oracle Database Enterprise Edition, включающая дополнительные возможности по обработке пространственных данных для поддержки ГИС-приложений, пространственных сервисов (location-based services), предназначенных для обработки и/или предоставления информации о местонахождении объектов.



постоянный характер обновления пространственных данных. При этом измененные пространственные данные записываются в БД Oracle Spatial.

Технология хранения пространственных данных не поддерживает механизм исключения дублирования геометрических объектов, что приводит к появлению в базе данных дублирующихся записей или объектов с одинаковой геометрией [24–27]. Несмотря на то что все пространственные объекты в базе данных имеют свой уникальный код, это не препятствует появлению в хранилище объектов с одинаковой геометрией, но разными идентификаторами. Это происходит в силу того что пространственные объекты часто копируются и вставляются, при этом встроенные механизмы геоинформационных систем, работающих с базой данных, не позволяют отследить объекты, имеющие одинаковую геометрию. Кроме того, дубликаты объектов появляются в результате неверных действий пользователей при обновлении пространственных данных. Например, объекты одного слоя могут быть дважды записаны в базу данных из файла чертежа. В таком случае загруженные из чертежа данные записываются «поверх» новых без их удаления.

Обычно процесс обновления данных в пространственном хранилище производится следующим образом:

- 1) удаляются все объекты требуемого слоя из базы данных;
- 2) из файла чертежа объекты вновь загружаются, включая новые или измененные;
- 3) после считывания все графические объекты записываются в БД;
- 4) запускается процедура генерации атрибутивных описаний объектов (информационных карточек) на основе графических объектов и их объектных данных. Таким образом, для каждого графического объекта создается своя собственная информационная карточка (описание), которая в дальнейшем может быть заполнена значениями свойств объекта.

Во всех современных ГИС к геометрическим объектам прикрепляются атрибутивные описания, характеризующие их свойства. Во многих ГИС атрибутивные описания хранятся непосредственно в таблицах вместе с геометрией объекта. Такой подход оправдан в случае узкоспециализированной ГИС, ориентированной на картографирование и управление определенным видом инженерных сетей. В прикладной ГИС универсального назначения атрибутивные описания хранят отдельно от графической информации, что позволяет в любой момент времени модифицировать их структуру согласно изменениям в инфраструктуре предприятия.

Объект атрибутивной схемы данных представляет собой совокупность характеризующих его базовых и дополнительных свойств.

В то время как пространственные объекты удаляются, соответствующие им атрибутивные описания остаются в базе данных до тех пор, пока они явно не будут удалены администратором системы. При генерации информационных карточек каждый раз происходит их повторное создание, вместо того чтобы найденные старые карточки перепривязать к тем же самым графическим объектам. Процесс перепривязки карточек затруднен в связи с тем обстоятельством, что привязка графических и атрибутивных объектов осуществляется исключительно по идентификатору графического объекта, который при повторной записи такого объекта в базу данных будет уже другим, и соответственно привязанная к нему информационная карточка будет «потерянной», но по-прежнему будет существовать в базе данных. Таким образом, выявляются две существенные проблемы: проблема поиска и удаления объектов, имеющих одинаковую геометрию, и проблема потери информационных карточек графических объектов при изменении у них идентификатора.

Технология хеширования пространственных данных позволяет найти и устранить дублирование одинаковых графических объектов и предотвратить генерацию избыточных атрибутивных описаний. Кроме того, «потерянные» информационные карточки всегда можно прикрепить к соответствующим графическим объектам даже при изменении у них идентификаторов.

Суть технологии состоит в следующем. Структура хранения атрибутивных и пространственных данных модифицирована таким образом, что помимо графической информации каждый пространственный объект имеет уникальный хеш-идентификатор, однозначно определяющий данный объект [28].

Множество всех графических объектов  $G$  может быть описано выражением ( $G = \{g_1, g_2, \dots, g_n\}$ ). В пространственной схеме каждый графический объект состоит из множества точек  $P$  и может быть представлен выражением  $g_i \sim P_i$ ,  $P_i = \{p_1, p_2, \dots, p_m\}$ .

Обозначим через  $F$  множество пространственных объектов, имеющих как минимум одно описание, а через  $N$  — множество графических объектов, не имеющих атрибутивных описаний. Таким образом, все множество пространственных объектов является объединением двух множеств ( $G = FUN$ ).

Множество всех атрибутивных объектов  $A$  описывается выражением  $A = \{a_1, a_2, \dots, a_k\}$ . Объект атрибутивной схемы можно определить выражением  $a_i = \{B, h\}$ , где  $B = \{b_1, b_2, \dots, b_l\}$  — множество свойств объекта;  $h$  — уникальный хеш-идентификатор.

Хеш-идентификатор представляет собой 32-байтный шестнадцатеричный код, вычисляемый алгоритмом MD5 [29] (Ronald L. Rivest, Massachusetts Institute of Technology Laboratory for Computer Science and RSA Data Security, Inc. April 1992, Request for Comments 1321), которому в качестве параметров передаются координаты всех точек графического объекта.

Функция хеширования совершает обход всех точек множества  $P_i$ , из которых состоит объект, и по совокупному набору точек рассчитывает уникальный хеш-код  $h$ . При этом используется алгоритм **MD5**, входным параметром которого является множество координат объекта. Функция хеширования, примененная к множеству точек объекта, осуществляет отображение множества координат точек объекта  $P_i$  в элемент  $h$ .

Хеширование обладает следующими свойствами:

1) функция осуществляет отображение множества точек объектов в уникальных хеш-кодах во множество из одного элемента  $f(N_i) = H_i$ , где  $N$  — множество графических объектов, не имеющих атрибутивных описаний;  $H$  — множество уникальных хеш-идентификаторов;

2) не существует обратной функции, способной по уникальному хеш-коду получить исходные данные. В таком случае координатами объекта являются  $F^{-1}(H) \neq N$  или  $F^{-1}(F(N)) \neq N$ ;

3) трудоемкость процесса нахождения двух сообщений, которые дают два одинаковых  $h$ -значения.

Таким образом, из третьего свойства следует, что если два графических объекта имеют одинаковый хеш-код, то это означает, что у них одинаковая геометрия, т. е. один из объектов является дубликатом второго. Технология использования хеш-кодов представлена на рис. 3.10.

Соответственно после обновления пространственных данных необходимо произвести выполнение процедуры, вычисляющей хеш-код, последовательно обходя все точки, из которых состоит объект. Рассчитанный по алгоритму MD5 хеш-код записывается в базу данных в ту же таблицу, где хранится геометрия объекта.

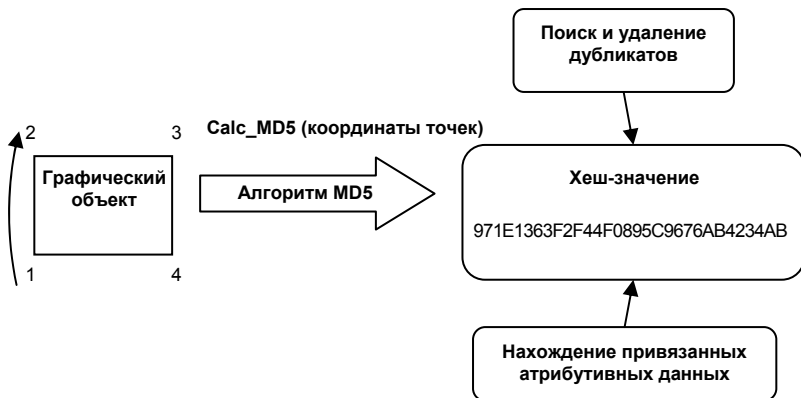


Рис. 3.10. Технология хеширования пространственных данных

Выполнение привязки пространственных данных к атрибутивным описаниям производится по требованию администратора и осуществляется вызовом специальной сервисной процедуры. Ниже приведен алгоритм работы процедуры привязки пространственных данных. Для каждого графического объекта  $g_i$  необходимо выполнить следующие шаги:

1) определить, принадлежит ли графический объект множеству  $N$ . Если да, то перейти к шагу 2, иначе — к шагу 7;

2) получить хеш-значение  $h_i$  графического объекта и выполнить проверку наличия во множестве  $H$  такого же хеш-идентификатора. Если хеш-значение  $h_i$  было найдено во множестве  $H$ , перейти к шагу 3, иначе — к шагу 5;

3) из множества атрибутивных объектов  $A$  получить уникальный идентификатор  $ID$  объекта  $a_{id}$ , найденный по его хеш-значению;

4) выполнить привязку графического объекта  $g_i$  к соответствующему атрибутивному объекту  $a_{id}$ , используя идентификатор  $ID$ , найденный на шаге 3. Перейти к шагу 7;

5) сгенерировать новое атрибутивное описание (объект  $a_{id}$ );

6) выполнить привязку графического объекта к сгенерированному на шаге 5 новому атрибутивному описанию;

7) выход.

Из множества геометрических объектов, не имеющих привязки, выполняется проверка существования атрибутивных объектов. Проверка производится по уникальному хеш-коду графического объекта. Таким образом, если во множестве атрибутивных описаний объектов находится объект, имеющий точно такой же хеш-идентификатор, то далее производится привязка этого графического объекта к его атрибутивному описанию. В противном случае для данного графического объекта вызывается функция генерации нового атрибутивного описания, после чего выполняется привязка только что созданного атрибутивного описания к графическому объекту.

Благодаря использованию хранимого хеш-значения каждого графического объекта процедура нахождения дубликатов занимает на порядок меньше времени, чем обычные запросы на полную проверку геометрии объектов. Кроме того, благодаря двойной привязке атрибутивных описаний к графическим объектам становится возможным нахождение «потерянных» атрибутивных описаний объектов и прикрепление их к соответствующим графическим объектам.

### **3.4. Технология разграничения прав доступа к векторным объектам**

В СУБД Oracle реализован объектный подход к хранению пространственных данных по принципу «один объект — одна запись». Помимо высокого быстродействия при работе с едиными хранилищами данных, построенными по объектному принципу, есть еще одно существенное преимущество: сложные аналитические запросы с пространственными составляющими могут выполняться не инструментальной ГИС, а средствами самой СУБД Oracle, что оптимально с точки зрения распределения ресурсов.

Единое хранилище на основе СУБД Oracle обладает способностью к репликациям, т. е. объединению данных из многих малых хранилищ в одно централизованное для целей глобального анализа, поддерживает параллельную индексацию и сегментирование таблиц. При этом вся техника поиска дубликатов записей, решения вопроса о дополнении, замене или удалении записей реализуется на основе встроенных механизмов самой СУБД.

Сегодня<sup>1</sup> при использовании СУБД Oracle атрибутивные и графические данные хранятся в одном месте, что увеличивает скорость и удобство доступа к ним. К тому же Oracle Spatial обеспечивает пользователям открытый доступ ко всем пространственным данным вне зависимости от того, хранятся ли данные в виде объектов в СУБД или в виде набора файлов на диске. Схема хранения данных и набор функций Oracle Spatial упрощают реализацию предоставления доступа и модификацию хранимых данных, повышают эффективность выполнения запросов. Кроме того, при использовании Oracle Spatial не возникает необходимости проверять связи между графической и атрибутивной частями ГИС-данных, так как эти части представляют собой одно целое [30].

При реализации веб-ориентированной ГИС в системе все данные хранятся на одном общем сервере, пользователи имеют доступ к этим данным посредством технологии Интернет. Это позволяет реализовать подход, при котором у пользователя какого-либо подразделения не появляются данные, которые не входят в круг его служебных обязанностей. Реализация этого подхода использует механизм детального контроля доступа к данным СУБД Oracle.

**Механизм детального контроля доступа** позволяет во время работы динамически присоединить предикат (предложения WHERE — где) как к одному, так и ко всем запросам к таблице или представлению. Таким образом, появилась возможность процедурной модификации запроса в процессе выполнения. Можно вычислить, *кто* выполняет запрос, *откуда* запрос выполняется, *когда* началось выполнение запроса, и сформировать предикат, соответствующий этим критериям. При использовании *Контекстов Приложения* можно незаметно через окружение (например, через роль, назначенную пользователю приложения) добавить дополнительную информацию и получить доступ к ней через процедуру или предикат.

Примером детального контроля доступа может служить политика безопасности, которая определяет, какие строки могут быть доступны различным группам пользователей. Политика безопасности формирует

---

<sup>1</sup> Раньше (когда-то единственно возможная, а теперь по многим причинам неэффективная) использовалась схема хранения ГИС-данных, при которой графика и семантика хранятся отдельно (в разных файлах). Это обстоятельство существенно снижает продуктивность ГИС-проектов, предназначенных для распределенного коллективного использования.

предикат, вид которого зависит от соединенного с базой пользователя и группы, к которой он принадлежит.

В случае с пространственными данными, когда пользователь активизируется, происходит запрос к базе данных, причем в зависимости от того, какую роль имеет пользователь, в запросе к базе данных подставляется предикатное предложение WHERE, которое налагает «фильтр» на значения в таблице, и в результате пользователь получает доступную только для него информацию.

Например, в предложении WHERE можно указать слои с графическими объектами, к которым пользователь имеет права доступа.

Для того чтобы работал описанный выше механизм, необходимо добавить процедуру политики безопасности. Эта процедура обращается к системному пакету DBMS\_RLS, в котором с помощью процедуры ADD\_POLICY можно добавить политику безопасности для пользователей ГИС.

Политики безопасности работают с ролями пользователей, соответственно необходимо раздать роли всем пользователям, которые работают под схемой базы данных. Через политику безопасности и механизм ролей пользователей можно реализовать контроль над функциями на добавление, удаление, изменение, просмотр информации по объектам геоинформационной системы, управляющей инженерными сетями.

DML<sup>1</sup>-предложения, обращающиеся к таблице, получают предикат, возвращаемый связанной функцией my\_security\_function независимо от источника, вызвавшего DML-операцию (т. е. независимо от приложения, получающего доступ к данным).

Использование механизма детального контроля доступа к данным СУБД Oracle обусловлено следующими причинами:

1) **возможностью легкой и простой поддержки.** Детальный контроль доступа позволяет иметь только одну таблицу и одну хранимую управляющую процедуру, которые заменят использование множества представлений. Создание множества представлений обычно приводит к увеличению числа объектов базы данных, так как для каждой группы пользователей требуется создание отдельного представления;

---

<sup>1</sup> DML (data-manipulation-language) — язык манипулирования данными. Примеры DML-предложений: SELECT, INSERT, UPDATE и DELETE.

2) **возможностью организации работы механизма на сервере.** Учитывая сложность управления и поддержки большого количества представлений, разработчики раз за разом стремятся закладывать логику приложения в самое приложение. Приложения просматривают, кто присоединен к базе данных, что он запрашивает, и выполняют соответствующий запрос. Это защищает данные, но только в случае, если доступ к ним осуществляется через данное приложение. Это снижает возможность использования средств выполнения запросов и генерации отчетов, а также других средств обработки данных. Повышается также вероятность получения искаженных данных, так как для того чтобы сделать искажение, достаточно подключиться к базе данных через любое другое средство, отличное от рассматриваемого приложения, и запросить данные. Благодаря включению в базу данных *логики безопасности*, то есть механизма, который определяет, какие данные может видеть пользователь, можно быть уверенным, что данные будут защищены независимо от используемого средства доступа к ним и обращение можно осуществлять с помощью любого средства, из которого возможен доступ к данным;

3) **возможностью запрета на соединение с базой данных от имени обобщенных пользователей.** Благодаря детальному контролю доступа каждый пользователь должен соединяться с базой данных под своим именем. В этом случае обеспечивается полная подотчетность — можно отслеживать действия на уровне пользователя. В прошлом многие приложения при работе с различными представлениями данных для различных пользователей должны были применять обобщенных пользователей базы данных соответственно выбираемым данным;

4) **упрощением разработки приложения.** Детальный контроль доступа забирает логику безопасности из логики приложения. Для поддержки безопасности данных разработчик приложения может сконцентрироваться на самом приложении, а не на логике низкоуровневого доступа к данным. Так как детальный контроль доступа полностью осуществляется на сервере, то приложения непосредственно наследуют эту логику. Раньше разработчики приложения должны были встраивать логику в приложение, делая приложение все более сложным для разработки и особенно сложным для его последующей поддержки. Если из приложения возможен доступ к данным, причем к одним и тем же данным и из нескольких точек приложения, то простейшее изменение политики



безопасности может затронуть множество модулей приложения. Благодаря применению детального контроля доступа изменения в политике безопасности не влияют на модули приложения;

5) *применением развитых средств разработки приложения.*

Во многих средах политика безопасности еще должным образом не определена и через некоторое время может измениться. Если происходит слияние компаний или другие структурные перемены или вводятся правила секретности, то политику безопасности необходимо изменить. Благодаря тому что управление доступом осуществляется на уровне, близком к данным, можно создать условия для развития приложения с минимальным влиянием и на приложение, и на средства разработки. Это является одной из причин для перехода к автоматическому использованию как новой логики, так и всех приложений и инструментов, позволяющих осуществлять доступ к базе данных со встроеной новой логикой.

Технология разграничения прав доступа к векторным объектам может быть положена в основу не только геоинформационных систем управления инженерными сетями, но и базовой концепции при построении любых других крупных информационных систем, содержащих географическую информацию на базе СУБД Oracle.

## АРХИТЕКТУРЫ ВЕБ-ОРИЕНТИРОВАННОЙ ГИС

### 4.1. Многоуровневая архитектура веб-ориентированных ГИС

Наряду с высокой потребностью публикации геопространственных данных в Интернете при построении комплексных ГИС возникает проблема создания наиболее эффективной архитектуры, которая должна обеспечить производительность, масштабируемость и надежность решения. В широком смысле построение архитектуры сводится к выбору основных составляющих системы: базовой геоинформационной технологии, средства хранения пространственных данных и т. д. В узком смысле — это применение наиболее эффективных архитектурных решений на каждом уровне проектируемой геоинформационной системы, в которой центральное место занимают средства веб-публикации.

В основе большинства современных ГИС лежит многоуровневая архитектура. В таких системах обычно выделяется три уровня: *обработка данных*, *бизнес-логика* и *представление*. В веб-ориентированных ГИС, как и в большинстве масштабных веб-приложений, можно условно выделить две части — клиентскую (Front-End) и серверную (Back-End), каждая из которых может иметь более сложную организацию и, оставаясь в рамках архитектуры, подразделяться на несколько уровней. Это, в свою очередь, требует решения задачи поиска наиболее эффективных подходов на каждом уровне проектируемой системы [31].

Разработка эффективной клиентской части (Front-End) — важный этап реализации средств веб-публикации геоинформационной системы, от результата которого зависит оценка качества и практичности приложения в целом. Сложность данного этапа обуславливается поиском наилучшего распределения бизнес-логики между клиентом и сервером и требованиями к функциональности: приложение должно предоставлять средства просмотра картографических данных, обеспечивать выполнение пространственных запросов и вывод атрибутивной информации. Опыт показывает, что реализация Front-End в веб-ориентированных ГИС — нетривиальная задача, требующая тщательного анализа.

Насыщение клиентского приложения богатой функциональностью порождает новый класс клиентских приложений, получивший название «богатый клиент», который, в отличие от «тонкого клиента», помимо уровня представления получает уровень бизнес-логики.

Применение архитектурных шаблонов, предназначенных для организации приложений, упрощает процесс разработки и сокращает время, затраченное на процесс создания веб-приложения. Одним из наиболее используемых является шаблон MVC (Model-View-Controller) (рис. 4.1).

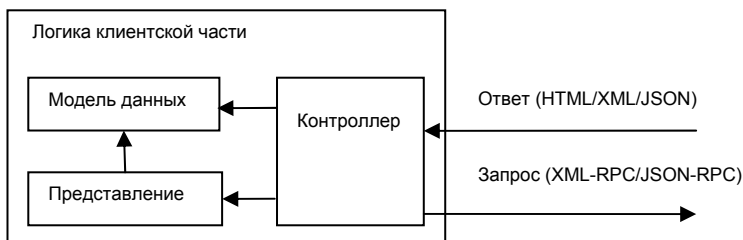


Рис. 4.1. Архитектура Front-End на основе модели MVC

Применение архитектуры MVC «в чистом виде» для разработки клиентской части (Front-End) имеет существенный недостаток: акцент в данной архитектуре делается на жесткое разделение ответственности и минимизацию кода, в результате чего затрудняется разработка многократно используемых однотипных компонентов, из которых состоит интерфейс геоинформационных систем (например, форм, таблиц, кнопок, картографического вьювера, элементов управления картой, поиском и пр.).

Применение технологий Веб 2.0 все более широким кругом пользователей обусловило неизбежное совершенствование подходов к созданию веб-приложений: применение асинхронной модели взаимодействия (AJAX) позволило повысить эффективность пользовательского интерфейса, а использование сервис-ориентированной архитектуры — оптимизировать передачу данных в процессе работы. В качестве средств интеграции различных веб-сервисов (Google Maps, Yahoo! Maps, Live Search Maps и др.) разрабатываются специализированные компоненты, к которым все чаще применяется термин «виджет» (widget — элемент интерфейса). Получить максимальную выгоду от использования этих подходов позволяет архитектура на основе компонентов.

Важным преимуществом компонентной архитектуры является повышение уровня абстракции при программировании пользовательских интерфейсов. Благодаря этому в распоряжение Front-End-разработчика предоставляются элементы, интерфейс которых не уступает компонентам графического интерфейса настольных систем. Для отображения таких компонентов необходимы динамическая генерация DOM-структур и механизмы управления каскадными стилями. В результате в среде браузера реализуются функции, типичные для настольных систем, и разработчик избавляется от необходимости программирования низкоуровневых операций.

## 4.2. Создание эффективного пользовательского интерфейса

Одной из причин низкой эффективности приложений, влияющих на степень интерактивности, является задержка между действием пользователя и реакцией на него. Непостоянное время отклика, связанное с задержками при передаче данных по сети, понижает практичность и затрудняет оценку качества приложения. Таким образом, низкое время отклика становится основным требованием, предъявляемым к пользовательскому интерфейсу. Существует несколько моделей взаимодействия пользователя и веб-приложения:

- 1) переходная модель (на основе синхронного взаимодействия);
- 2) независимая модель (на основе асинхронного взаимодействия).

**Переходная модель** является классической моделью веб-приложения, в основе которой лежит совокупность статических или динамических веб-страниц. Пользователь, выполняя различные действия, фактически осуществляет переход от одной страницы к другой и тем самым вынужден синхронизировать свои действия с сигналами об успешном завершении той или иной операции. Ввиду изначально присущих приложению недостатков, связанных с получением отклика, приложение, отвечающее переходной модели, не может завладеть всем вниманием пользователя. На время выполнения каждой из функций деятельность пользователя приостанавливается. На рис. 4.2 представлена диаграмма последовательности процесса использования картографических данных в синхронном режиме.

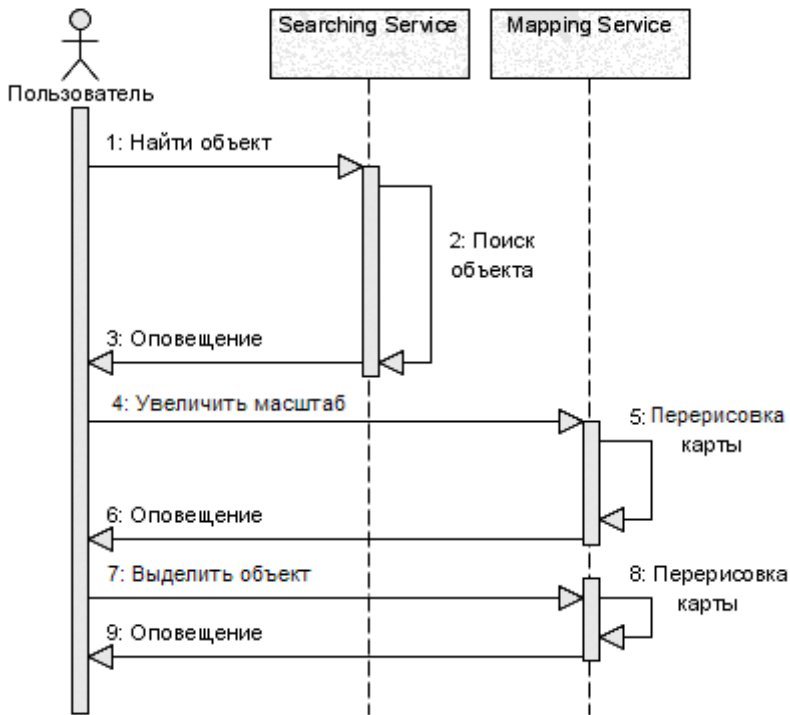


Рис. 4.2. Синхронное взаимодействие пользователя и ГИС

**Независимая модель** благодаря принципам асинхронного взаимодействия освобождена от недостатков переходной модели. Каждая операция считается длительной и выполняется асинхронно в отдельном потоке (в фоновом режиме), а пользователь в это время может выполнять другие действия. При этом время отклика сводится к минимуму, уменьшается влияние задержек, связанных с длительными вычислениями и взаимодействием по сети. На рис. 4.3 представлена диаграмма последовательности процесса использования картографических данных в асинхронном режиме. При асинхронном взаимодействии пользователь, активизируя функцию поиска объекта и не дожидаясь оповещения об успешном выполнении операции, может продолжить использование других функций.

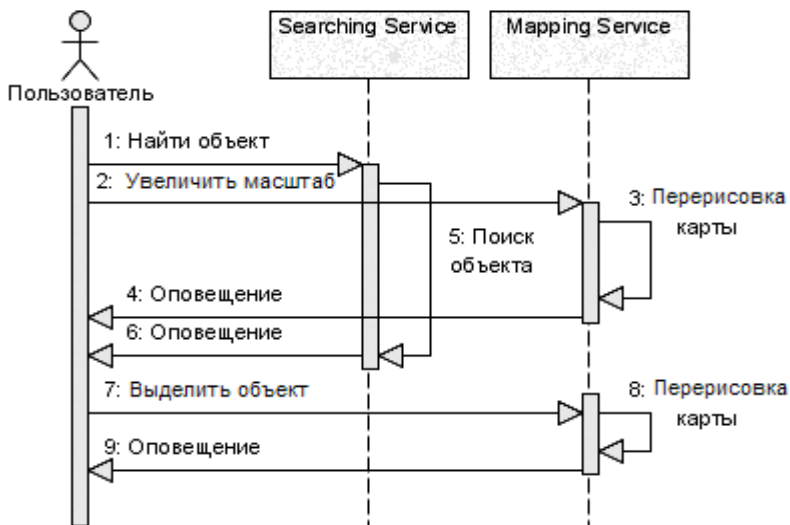


Рис. 4.3. Асинхронное взаимодействие пользователя и ГИС

### ***Жизненный цикл независимого и переходного приложений***

Для классического приложения на основе переходной модели браузера представляет собой низкоуровневый терминал. Он не имеет информации об этапе выполнения работы пользователем, а на сервере содержатся минимальные сведения, которые сводятся к поддержке сеанса.

Когда пользователь инициирует сеанс, на стороне сервера инициализируется пользовательская модель данных. Сервером формируется поток HTML-данных, после чего исходная страница доставляется браузеру. При каждом обращении к серверу браузер получает очередную страницу, содержащую данные тех же типов, что в предыдущих документах. Когда пользователь завершает сеанс или закрывает браузер, выполнение приложения прекращается и сеанс оканчивается.

В независимом приложении часть прикладной логики переносится на клиента. После инициализации сеанса клиентское приложение доставляется браузеру. Это приложение способно реагировать на действия пользователя самостоятельно. Если имеющихся в наличии возможностей не хватает, оно передает запросы серверу, не прерывая последовательность действия пользователя.

Анализ моделей взаимодействия пользователя и веб-приложения как подходов к построению пользовательского интерфейса позволяет сделать вывод о том, что создание клиентской части веб-ориентированной геоинформационной системы должно базироваться на использовании независимой модели взаимодействия. Возможными средствами реализации могут стать такие технологии, как AJAX<sup>1</sup>, Adobe Flash<sup>2</sup>, Java Web Start<sup>3</sup>, .NET No Touch Deployment<sup>4</sup> и пр. Применение подхода AJAX позволяет создавать более эффективные решения, так как, в отличие от других средств, AJAX поддерживается большинством браузеров и не нуждается в заранее установленной исполняющей системе.

### 4.3. Выбор архитектуры приложения на стороне сервера

Одним из ключевых этапов проектирования приложения является процесс разделения функциональности на несколько уровней. Эта задача решается посредством применения различных многоуровневых архитектур. После логического расслоения возникает задача физического распределения уровней приложения.

---

<sup>1</sup> AJAX (Asynchronous Javascript and XML) — подход к построению интерактивных пользовательских интерфейсов веб-приложений, заключающийся в «фоновом» обмене данными браузера с веб-сервером. В результате при обновлении данных веб-страница не перезагружается полностью и веб-приложения становятся более быстрыми и удобными.

<sup>2</sup> Adobe Flash — мультимедийная платформа компании Adobe для создания веб-приложений. Широко используется для создания рекламных баннеров, анимации, игр, а также воспроизведения на веб-страницах видео- и аудиозаписей.

<sup>3</sup> Java Web Start (часто JavaWS) — технология компании Sun Microsystems, позволяющая запускать приложения на Java из браузера. Основана на протоколе Java Network Launching Protocol (JNLP). В отличие от апплетов, приложения Web Start запускаются не в окне браузера и не имеют с ним прямой связи.

<sup>4</sup> .NET No Touch Deployment («развертывание без контакта») — технология развертывания Windows-приложений. Пользователи могут «рассматривать» исполняемые файлы Windows, как если бы они были обычными веб-страницами, без всякой инсталляции.

В классических веб-приложениях уровень представления доставляется клиенту, а остальные уровни находятся на сервере. В независимых приложениях уровень бизнес-логики распределен между клиентом и сервером, поэтому для функционирования распределенного приложения сервер должен обеспечивать доставку клиентской части приложения браузеру.

В вариантах архитектур, предназначенных для построения классических веб-приложений, с учетом достоинств и недостатков каждого подхода можно выделить наиболее существенные детали, использование которых позволит создать оптимальную серверную архитектуру веб-ориентированной геоинформационной системы.

**Архитектура Model2** является разновидностью шаблона проектирования MVC. Как и MVC, данная архитектура разделяет модель данных приложения, пользовательский интерфейс и управляющую логику на три отдельных компонента. При этом модификация одного из компонентов оказывает минимальное воздействие на другие компоненты.

Отличие архитектуры Model2 от MVC в том, что контроллер имеет единственную точку входа и единственное определение последовательности действий пользователя. Данное решение, именуемое «Контроллер запросов» (Front Controller), объединяет все действия пользователя по обработке запросов в одном месте, распределяя их выполнение посредством одного объекта-обработчика (рис. 4.4).

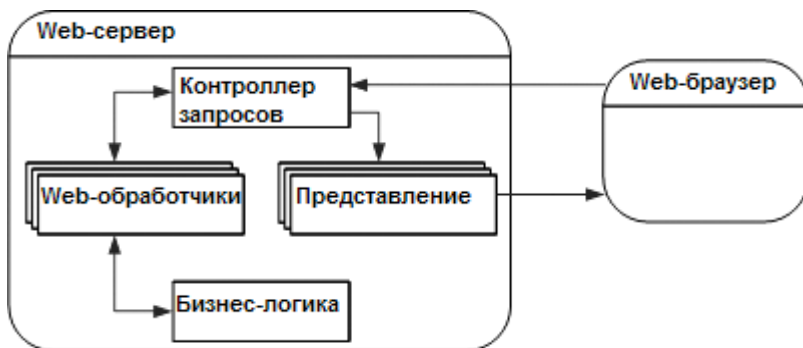


Рис. 4.4. Архитектура Model2



Из каждого запроса, поступающего к веб-серверу, извлекается название веб-обработчика и иерархия команд. Веб-обработчик представляет собой объект, который выполняет фактическое получение POST-или GET-запросов, поступивших на веб-сервер. Контроллер запросов извлекает необходимую информацию из адреса URL и входных данных запроса, после чего решает, какое действие необходимо инициировать, и делегирует его выполнение соответствующей команде.

Данная архитектура хорошо подходит для доставки данных, но не определяет средств доставки клиентского приложения, что может быть существенным недостатком при разработке «богатой» клиентской части веб-ориентированной ГИС.

**Архитектура на основе компонентов** позволяет повысить уровень абстракции при программировании пользовательских интерфейсов (рис. 4.5). Благодаря этому в распоряжение разработчика серверных программ предоставляются элементы, интерфейс которых не уступает компонентам графического интерфейса для настольных систем.

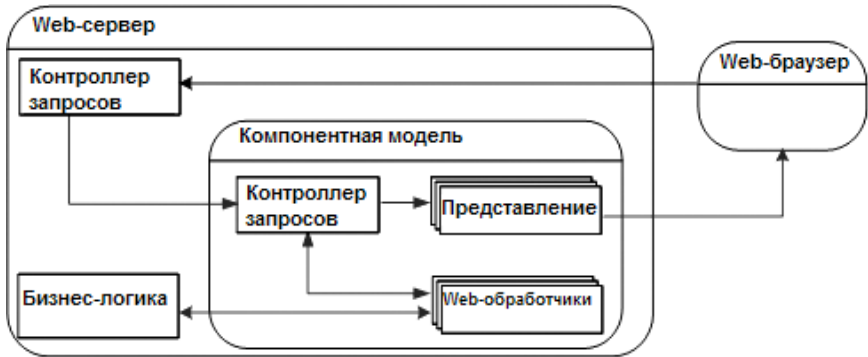


Рис. 4.5. Архитектура на основе компонентов

Для отображения таких компонентов необходима автоматическая генерация потока HTML-данных и JavaScript-программ. В результате в среде браузера реализуются функции, типичные для настольных систем, и разработчик избавляется от необходимости программировать низкоуровневые операции (см. рис. 4.5).

Преимуществом компонентного подхода является возможность реализации системы воспроизведения на базе обычного HTML-кода, в результате чего исчезает необходимость специального изучения разработчиками особенностей языка сценариев (например, JavaScript).

Применение компонентной архитектуры «в чистом виде» для создания серверной части ГИС имеет ряд недостатков:

1) подобное решение хорошо подходит лишь приложениям, для которых требуются только стандартные типы компонентов. Создание сложного приложения, такого как картографический вьювер, потребовало бы написания собственного набора компонентов, от прокручиваемой карты до элементов масштабирования и всплывающей подсказки;

2) при построении сложного приложения на основе независимой модели архитектура должна обеспечивать эффективную передачу данных в процессе работы, что достигается посредством механизмов обработки событий серверной частью приложения. Однако контроллер в данной архитектуре жестко привязан к уровню сервера, ориентирован на работу с компонентами и лишен гибкости для определения собственных обработчиков событий.

**Архитектура на основе веб-служб** ориентирована на работу с информацией, предоставляемой бизнес-логикой приложения. В данном случае служба — это сущность, к которой можно обратиться по сети и получить в качестве ответа структурированный документ. Программа, использующая веб-службу как источник данных, обладает высокой степенью автономности и более широкими возможностями повторного использования элементов приложения. Служба определяется один раз и применяется различными клиентами, работающими независимо друг от друга. Таким образом, данная архитектура разделяет средства, предназначенные для генерации клиентской части приложения и ее обслуживания.

Создание комплексной геоинформационной системы, интегрирующей множество технологий, может потребовать от серверной архитектуры большей гибкости. Например, при синхронизации модели данных клиента и сервера, как правило, используются механизмы обработки событий, а при использовании систем, требующих доставки клиентской части приложения, эффективна архитектура на основе контроллера запросов.

**Усовершенствованная архитектура на основе веб-служб с использованием развитого контроллера.** С учетом достоинств и недостатков вышеописанных архитектур выделяются основные требования к

серверной архитектуре веб-ориентированной ГИС: поддержка обработки событий; использование веб-служб; поддержка средств доставки клиентской части приложения. Удовлетворить данные требования позволяет усовершенствованная архитектура, разработанная на основе подхода MVC (рис. 4.6).

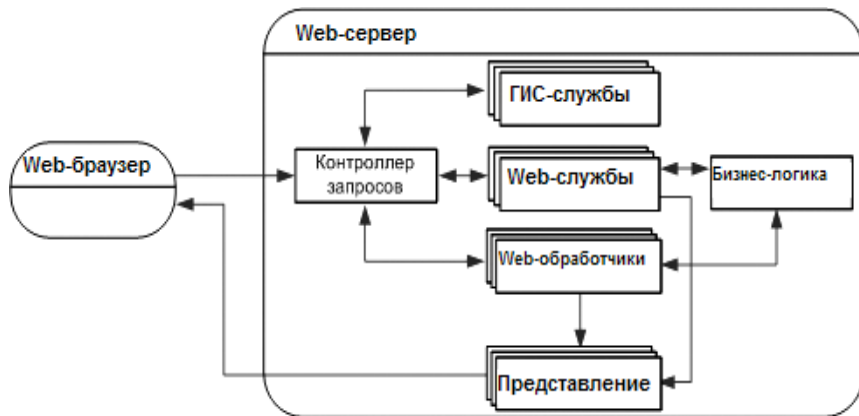


Рис. 4.6. Усовершенствованная архитектура на основе веб-служб

Организация уровня управляющей логики предполагает реализацию контроллера запросов в виде цепочки фильтров. Фильтры — это программные компоненты, позволяющие разбить процесс обработки запроса на отдельные этапы и выделить различные вспомогательные действия. Логика каждого фильтра реализована таким образом, что переход к последующему фильтру зависит от успешного выполнения действий предыдущего.

В предлагаемой архитектуре [32] запрос обрабатывается фильтрами в следующей последовательности (рис. 4.7):

1) инициализация — создание или открытие сеанса работы пользователя и обновления статистической информации о пользователе;

2) диспетчеризация запроса — анализ и фильтрация некорректных URL-запросов, выбор контроллера, иерархии команд, веб-обработчика и выделение запросов к ГИС-службам;

3) ГИС-обработчик — подпрограмма, которая получает запросы картографического вьювера, перенаправляет их соответствующим ГИС-службам и передает результат браузеру;

4) авторизация — проверка прав доступа на выполнение действий;

5) генерация HTTP-ответа — отправка данных и HTTP-заголовков;

6) обработка запроса — запуск команды, которую обрабатывает указанный контроллер и веб-обработчик;

7) генерация представления — формирование представления данных или содержимого в заданном формате и генерация клиентской части приложения.

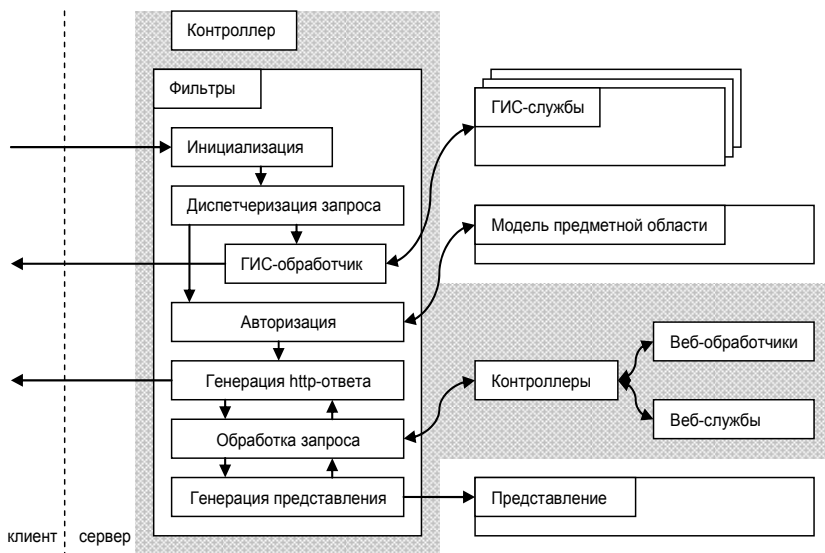


Рис. 4.7. Работа контроллера на основе фильтров

Данная архитектура обладает преимуществами архитектуры с использованием веб-служб и достоинствами применения подхода MVC. Применение развитого контроллера на базе фильтров позволяет управлять ходом работы приложения, более эффективно обрабатывать события и исключения, а также интегрировать в серверное приложение средства публикации картографических данных и ГИС-службы.

## 4.4. Выбор архитектуры приложения на стороне клиента

Проблема создания сложного и масштабируемого клиента обострилась с ростом популярности реализаций технологии асинхронного взаимодействия. Теперь к клиенту предъявляются иные требования: он должен обладать не только богатой функциональностью, но и интеллектуальностью реализуемой логики, что выражается в способности выполнять большинство операций вместо сервера и способности самостоятельно отвечать на действия пользователя. Приложение, способное удовлетворить данное требование, должно строиться на основе независимой модели взаимодействия и обеспечивать малое время отклика, где основную роль играет эффективность обмена данными между клиентом и сервером.

Данные, возвращаемые сервером, можно разделить на три типа:

1) *содержимое*. В качестве данных передается генерируемый сервером HTML-поток, отображаемый в составе элемента IFrame или встраиваемый в структуру документа средствами DHTML. Подход эффективен при передаче статических данных, например справочной информации;

2) *сценарии*. В качестве данных передаются JavaScript-сценарии, которые позволяют динамически генерировать представление либо содержат в себе структуры данных модели предметной области. Недостатком данного подхода является тесная связь между уровнями приложения, так как для генерации кода на стороне сервера необходимо иметь информацию о программном интерфейсе клиента;

3) *структурированные данные*. Сервер передает только низкоуровневые данные (XML<sup>1</sup>/JSON<sup>2</sup>), которые обрабатываются уровнем клиента

---

<sup>1</sup> XML (*eXtensible Markup Language*) — текстовый формат, предназначенный для хранения структурированных данных (взамен существующих файлов баз данных), обмена информацией между программами, а также создания на его основе более специализированных языков разметки.

<sup>2</sup> JSON (JavaScript Object Notation) — формат обмена данными, используемый для представления данных в бизнес-логику, выполняющуюся в браузерах. Многие Ajax-разработчики предпочитают обрабатывать данные напрямую, используя JSON в JavaScript-коде на стороне браузера. По мере расширения применения JSON в программах, работающих на серверах промежуточного уровня, станет необходимым предоставлять данные корпоративных приложений в браузеры в формате JSON, а не в XML-формате. Это означает, что разработчики должны преобразовать существующие корпоративные данные на стороне сервера, закодированные в XML-формате, в JSON-формат перед передачей их в браузер.

и используются для обновления клиентской модели данных и пользовательского интерфейса. При использовании данного подхода удастся разделить уровни сервера и клиента, что позволяет изменять код клиента и сервера независимо друг от друга.

При разработке веб-ориентированной инженерной ГИС, представленной в пятой главе данной работы, была использована архитектура (рис. 4.8), основанная на принципах MVC и ориентированная на использование структурированных данных, поставщиками которых являются веб- и ГИС-службы [32].

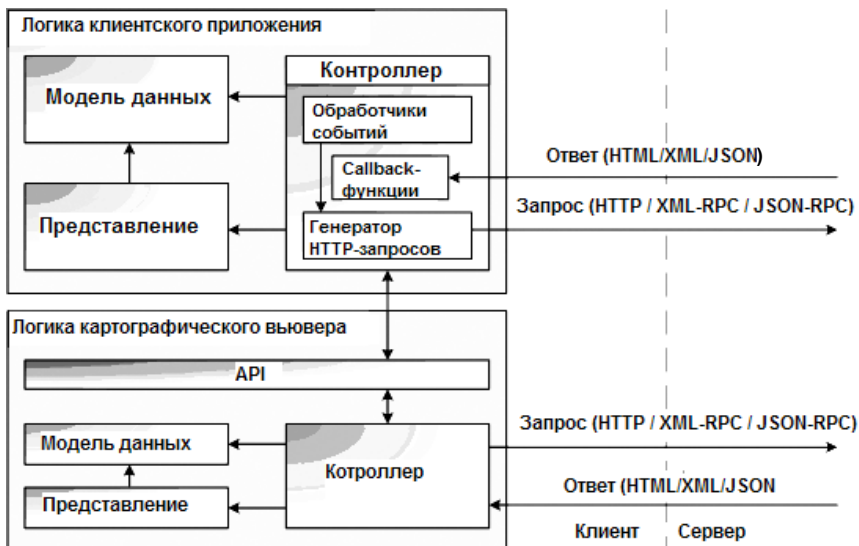


Рис. 4.8. Архитектура приложения на стороне клиента

Картографический вьювер, интегрированный в клиентское приложение, реализован в виде компонента, управление которым осуществляется посредством API-интерфейса. Контроллер в приложении включает в себя обработчики событий, перехватывающие действия пользователя, генератора запросов, подготавливающего данные к отправке на сервер, и набора Callback-функций, которые выполняют обновление модели данных и представления при получении асинхронного ответа.

Данная архитектура обладает следующими преимуществами [32]:

- 1) применение архитектуры MVC обеспечивает разделение ответственности и уменьшение количества избыточного кода;
- 2) использование независимой модели взаимодействия позволяет повысить интерактивность приложения;
- 3) ориентированность на структурированные данные позволяет разделить код клиента и сервера, а также повлиять на уменьшение времени отклика;
- 4) разделение приложения на отдельные компоненты способствует коллективной разработке приложения, помогая разработчикам сосредоточиться на выполнении функций, требующих определенной специализации, и взаимодействовать через четкие интерфейсы.

Данная архитектура была положена в основу инструментальной веб-ориентированной ГИС WGS3, которой посвящена пятая глава.

Эффективность предложенных архитектур была подтверждена в ходе их использования для обеспечения средств веб-публикации геоинформационных систем промышленных предприятий ООО «Томск-нефтехим» (г. Томск), ОАО «КМК» (г. Новокузнецк), ООО «Северский водоканал» (г. Северск), а также ГИС Томского университета систем управления и радиоэлектроники.

## **4.5. Сервис-ориентированный подход при построении геоинформационной системы**

Одной из наиболее важных задач прикладного уровня, используемых в геоинформационных системах, является задача пространственно-временного анализа накопленных данных. К сожалению, в современных веб-ориентированных ГИС данная задача ограничивается функциями измерения расстояния, построения буферных зон, нахождения маршрута между двумя заданными точками и просмотра тематических карт по имеющейся статистической информации.

Пространственно-временной анализ должен проводиться с целью выявления наличия, вида взаимосвязей и закономерностей в пространственном распределении отдельных объектов, классов объектов или их характеристик.

При проектировании архитектуры современной веб-ориентированной ГИС авторами был сделан выбор в пользу сервис-ориентированного подхода<sup>1</sup>.

Сервис-ориентированная архитектура предполагает разделение системы на компоненты, или «сервисы», которые, имея согласованные общие интерфейсы, используют единые правила для определения того, как вызывать сервисы и как они будут взаимодействовать друг с другом. Сервис (служба) — программный компонент, доступ к которому можно получить через компьютерную сеть. Он реализует некоторую функцию с четко определенным интерфейсом.

Наиболее часто к принципам SOA относят:

- 1) необязательность привязки архитектуры к какой-то определенной технологии;
- 2) независимость организации системы от используемых вычислительных платформ;
- 3) независимость организации системы от применяемых языков программирования;
- 4) использование сервисов, независимых от конкретных приложений, с единообразными интерфейсами доступа к ним;
- 5) организация сервисов как слабосвязанных компонентов для построения систем.

Преимуществами SOA являются:

- 1) сокращение издержек при разработке приложений за счет упорядочивания процесса разработки;
- 2) расширение повторного использования кода;
- 3) независимость от используемых платформ, инструментов, языков разработки;
- 4) повышение масштабируемости создаваемых систем;
- 5) улучшение управляемости создаваемых систем.

В рамках сервис-ориентированной архитектуры основу корпоративной ГИС предлагается формировать из комплекса специализированных служб (картографических сервисов) [32]. В качестве таких служб могут выступать сервисы пространственного анализа; сервисы публикации

---

<sup>1</sup> SOA (Service-oriented architecture) — модульный подход к разработке программного обеспечения, основанный на использовании сервисов (служб) со стандартизированными интерфейсами.



пространственных данных (генерация растрового или векторного представления); сервисы поиска информационных (атрибутивных) описаний; сервисы генерации отчетов на основе атрибутивных данных.

На сегодняшний день принципы SOA наиболее эффективно (применительно к веб-ориентированным системам) реализуют архитектурные стили (подходы) WSA (Веб Services Architecture) и REST (Representational State Transfer), которые предлагается рассмотреть в контексте построения сервис-ориентированной архитектуры корпоративной ГИС.

### WSA (Web Services Architecture)

Веб-сервисы в данной архитектуре базируются на широко распространенных и открытых протоколах: HTTP, XML, UDDI<sup>1</sup>, WSDL<sup>2</sup> и SOAP<sup>3</sup>. Эти протоколы реализуют основные требования SOA — сервис должен поддаваться динамическому обнаружению и вызову (UDDI, WSDL и SOAP), а также должен использоваться не зависящий от платформы интерфейс (XML). Протокол HTTP обеспечивает функциональную совместимость (рис. 4.9).

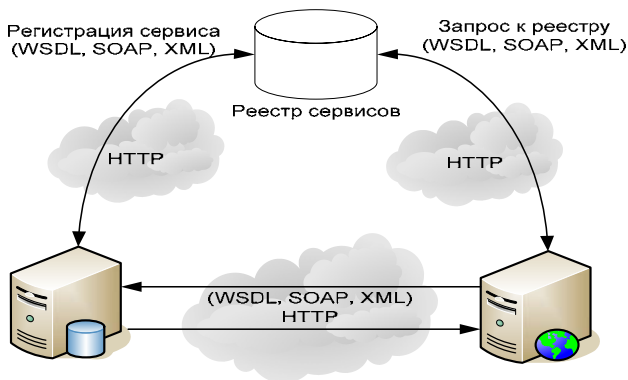


Рис. 4.9. Архитектура Web Services Architecture

<sup>1</sup> UDDI (Universal Description Discovery & Integration) — инструмент для расположения описаний веб-сервисов для последующего их поиска другими организациями и интеграции в свои системы.

<sup>2</sup> WSDL (Web Services Description Language) — язык описания веб-сервисов, основанный на языке XML.

<sup>3</sup> SOAP (Simple Object Access Protocol) — протокол, используемый для обмена произвольными сообщениями в формате XML.

WSA предоставляет возможность реализовать SOA, в которой службы взаимодействуют, обмениваясь XML-сообщениями. Технологии создания веб-служб в настоящее время достаточно продуманны, предлагают как коммерческие, так и свободно распространяемые стеки веб-служб. Для разработки картографических веб-сервисов существуют специальные инструментальные средства, благодаря чему разработчики могут реализовать предоставляемые ГИС-функции в виде SOAP веб-сервисов.

К достоинствам данной архитектуры можно отнести надежность (поддержку спецификации WS-Reliability<sup>1</sup>) и безопасность (поддержку спецификации WS-Security<sup>2</sup>). Однако при реализации приложения предполагается наличие инструментальных средств разработки (как правило, коммерческих), при отсутствии которых быстро создать простой и эффективный сервис — задача нетривиальная. Помимо этого, при реализации достаточно простых и часто используемых картографических функций (визуализации, запроса атрибутивной информации) сервис будет обладать значительным временем отклика и некоторой избыточностью пересылаемой информации, что при больших нагрузках может привести к перегруженности вычислительных ресурсов и сетевого трафика. Поэтому в качестве более простой реализации SOA часто рассматривают архитектурный подход REST.

### **REST (Representational State Transfer)**

В архитектуре REST (рис. 4.10) система представляется отдельными ресурсами со своим состоянием. Система не хранит состояние клиентского процесса для *масштабируемости*. Ресурсы представлены стандартными time-типами для независимости от платформы и самоописываемости, имеют универсальный интерфейс, идентифицируются собственными URL и связываются только через явное указание ссылок и форм в их представлениях для обеспечения *интеграции* между сервисами.

---

<sup>1</sup> WS-Reliability — спецификация для передачи сообщений открытых, надежных Web-сервисов, включающая гарантированную доставку, удаление дубликатов сообщений и их упорядочивание. Характеристики надежности основаны на расширениях протокола SOAP, а не привязаны к базовому транспортному протоколу. Благодаря данной спецификации широкий спектр систем сможет взаимодействовать независимо от платформы и производителя.

<sup>2</sup> WS-Security — спецификация, отвечающая за конфигурирование аутентификации благодаря использованию цифровых подписей. Построена как расширение протокола SOAP.

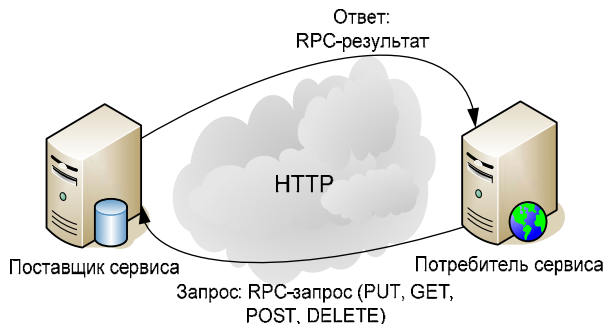


Рис. 4.10. Архитектура Representational State Transfer

Для доступа к ресурсу используется небольшое количество методов HTTP (GET, PUT, DELETE, POST). Передача метаданных осуществляется посредством HTTP-заголовков (mime-типов). За кеширование, проксирование и авторизацию также отвечает протокол HTTP.

Кардинальное отличие REST от SOAP-сервисов заключается в том, что в REST акцент сделан на доступ к ресурсам, а не на исполнение удаленных сервисов. Если SOAP-клиенты запрашивают выполнение действия на сервере, то REST-клиенты требуют сам ресурс. Например, вместо того чтобы запрашивать на сервере визуализацию фрагмента карты, запрашивается сам фрагмент карты как статическая картинка. По такому же принципу может быть осуществлен доступ к службам WMS (Web Map Service) и WFS (Web Feature Service).

Учитывая специфику REST и SOAP-сервисов, представляется наиболее эффективным распределить картографические сервисы исходя из интенсивности их использования, сложности исполняемой логики и требований к надежности процесса выполнения. SOAP-сервисы обладают более высокой надежностью за счет спецификаций безопасного доступа к службам (WS-Security), адресации служб (WS-Addressing), управления состоянием (WS-Resource Framework), но сильнее загружают вычислительные ресурсы, поскольку требуют разбора XML-кода и упорядочивания объектов, а для оптимизации управления и ускорения этих операций необходимо специальное программно-аппаратное обеспечение. REST-сервисы менее надежны, но более производительны и позволяют снизить затраты на обеспечение сервис-ориентированного доступа к ресурсам.

Таким образом, к SOAP-сервисам можно отнести службы:

- управление доступом;
- пространственный анализ;
- поиск по атрибутивным данным.

К REST-сервисам относятся следующие службы:

- визуализация;
- генерация KML<sup>1</sup>/GeoRSS<sup>2</sup>-представлений;
- сервисы WMS;
- сервисы WFS.

Анализ данных архитектурных подходов позволяет сделать вывод о том, что при оптимальном распределении функций между REST и SOAP-сервисами можно повысить эффективность применения SOA для реализации корпоративной ГИС. Это позволит снизить время выполнения критических для ГИС функций, уменьшить сетевой трафик, а также в некоторых случаях снизить время издержки на разработку картографических сервисов.

## 4.6. Построение веб-ориентированной ГИС как Mashup-приложения

Спроектированную с учетом приведенных принципов инфраструктуру картографических сервисов предлагается положить в основу архитектуры корпоративной веб-mashup<sup>3</sup>-ГИС, которая в рамках данной архитектуры будет являться провайдером содержимого (источником данных).

---

<sup>1</sup> KML (Keyhole Markup Language) — язык разметки на основе XML для представления трехмерных геопространственных данных в программе Google Earth, имевшейся Keyhole до ее приобретения Google.

<sup>2</sup> GeoRSS — развивающийся стандарт для встраивания информации о местоположении в новостные ленты. В стандарте GeoRSS информация о местоположении состоит из географических точек, линий и многоугольников, а также описаний соответствующих особенностей. GeoRSS-ленты разработаны с учетом возможности их использования в программном обеспечении, связанном с географией, таком как генераторы карт.

<sup>3</sup> Mashup (или mash-up) — приложение, комбинирующее в себе контент с различных источников («гибрид» веб-приложений).

**Mashup-приложение** объединяет данные из нескольких источников в один интегрированный инструмент. Например, в качестве подложки могут быть использованы картографические данные виртуальных глобусов (Google Maps, Virtual Earth), в качестве тематических слоев — поток KML-данных, предоставленный специализированным картографическим сервисом, а для добавления к ним произвольных атрибутивных данных необходимо обратиться к сервисам управления информационными описаниями. В результате создается новый уникальный веб-сервис, изначально не предлагаемый ни одним из источников.

Архитектура mashup-приложений всегда включает три части:

1) *провайдер содержимого* — источник данных. Данные доступны через API и различные веб-протоколы, такие как RSS, REST и веб-сервисы;

2) *mashup-сайт* — веб-приложение, предлагающее новый сервис, использующий не принадлежащие ему источники данных;

3) *браузер клиента* — пользовательский интерфейс mashup-приложения. В веб-приложениях содержимое может быть «смешано» клиентским браузером с использованием клиентского языка программирования, например JavaScript.

Основу mashup-приложения составляет комплекс компонентов — виджетов (Widgets), представляющих собой небольшие программные модули, имеющие графический интерфейс и прикладной интерфейс программирования (API), которые поддерживают систему событий. Основная задача виджета — интегрировать веб-сервис в приложение (выполнить смешивание). Интеграция картографических сервисов посредством виджетов представляет наиболее эффективный подход к реализации веб-ориентированной корпоративной ГИС в рамках SOA, поэтому архитектура mashup-приложений требует более детального рассмотрения.

Если рассматривать архитектуру mashup-приложения на прикладном уровне, то это шаблон проектирования «Facade», который предлагает упрощенный интерфейс для большего кода (в этом случае кода для агрегации различных сервисов с различными API).

На уровне взаимодействия сервисов различаются два типа архитектур mashup-приложений: на основе серверного смешения (Server-Side) и клиентского смешения (Client-Side). В соответствии с этим авторами предлагается два типа архитектур веб-mashup-ГИС.

### Архитектура сервис-ориентированной веб-ГИС на основе серверного смешивания (mashup)

События, генерируемые интерфейсом приложения в ответ на действия пользователя, обрабатываются логикой клиентского приложения (слой «контроллер» паттерна MVC), которая, в свою очередь, посылает HTTP-запросы веб-ГИС-серверу (рис. 4.11).

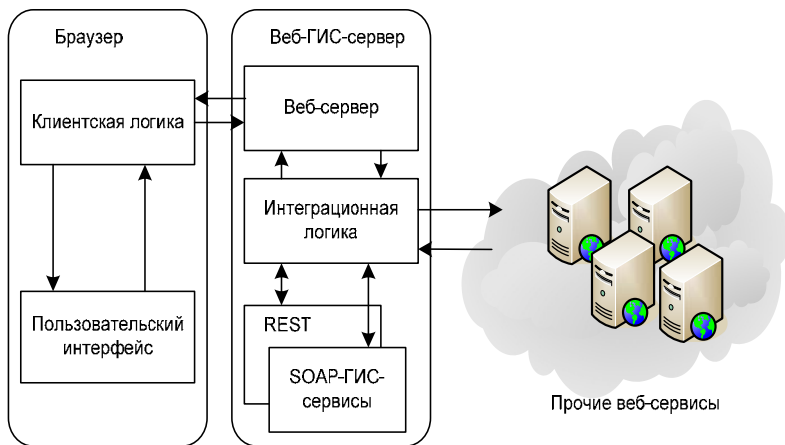


Рис. 4.11. Архитектура на основе серверного смешивания

Веб-сервер (начальное звено веб-ГИС-сервера) инициирует работу интеграционной логики, которая посредством API серверных компонентов интегрирует как внутренние сервисы (ГИС), так и любые внешние веб-сервисы.

### Архитектура сервис-ориентированной веб-ГИС на основе клиентского смешивания (mashup)

В отличие от архитектуры на основе серверного смешивания, в данной архитектуре интеграционная логика размещается на клиенте. Виджеты, интегрированные в пользовательский интерфейс, генерируют события, которые могут быть обработаны как клиентской логикой, так и самими компонентами, и могут взаимодействовать между собой посредством API. Виджеты работают напрямую с ГИС-сервисами и со сторонними веб-сервисами посредством REST и SOAP-запросов (рис. 4.12).

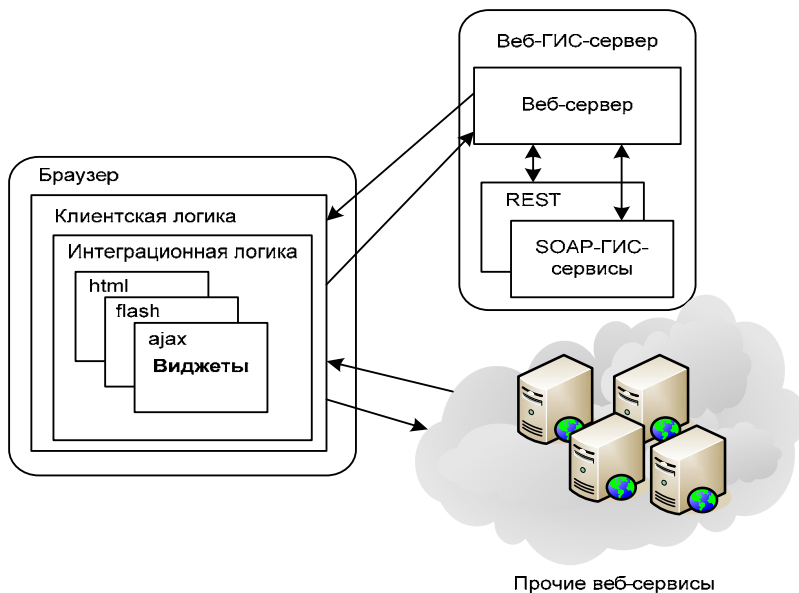


Рис. 4.12. Архитектура на основе клиентского смешивания

Представленные архитектуры позволяют перенести идеологию смешивания веб-сервисов на сектор веб-ориентированных геоинформационных систем, позволяя тем самым повысить эффективность применения SOA за счет реализации основных преимуществ mashup-приложений:

- повышения гибкости разработки, позволяющей быстро собирать и конфигурировать приложения;
- ускорения реализации и снижения стоимости работ за счет легкой интеграции, возможности многократного использования и заимствования;
- построения SOA как бизнес-ориентированной и явной системы путем расширения возможности многократного использования сервисов и виджетов;
- стимулирования внедрения новых технологий и идей при сохранении соответствующего уровня управляемости.

## 4.7. Сервис-ориентированная архитектура блока анализа веб-ориентированной ГИС

В рамках концепции SOA были рассмотрены функции пространственного анализа, расширяющие функциональные возможности геоинформационной системы и представленные в виде отдельных сервисов [32]. На рис. 4.13 представлена сервис-ориентированная архитектура блока анализа веб-ориентированной ГИС.



Рис. 4.13. Сервис-ориентированная архитектура блока анализа веб-ориентированной ГИС



В представленной архитектуре сервисы объединяются в группы по способу реализации и специфике анализа:

- сервисы подготовки данных для анализа;
- базовые сервисы анализа;
- специализированные сервисы анализа;
- сервисы представления результатов анализа.

### **Сервисы подготовки данных для анализа**

Для получения качественных результатов проводимого анализа необходимо правильно обработать и подготовить пространственные данные. Начальным этапом подготовки данных для использования их в веб-ориентированных геоинформационных системах является извлечение знаний о пространственном распределении огромного набора географических объектов. Сервис *spatial data mining* (пространственное извлечение знаний) решает данную задачу, используя инструментарий *spatial data mining*. Извлечение знаний — это процесс обнаружения в «сырых» данных ранее неизвестных, нетривиальных, практически полезных и доступных интерпретации знаний или шаблонов, необходимых для принятия решений в различных сферах человеческой деятельности.

Различие между классическим и пространственным извлечением состоит в том, что в классическом статистическом анализе элементы выборки получаются независимо, а в пространственном анализе предположение о независимости становится неприемлемым из-за наличия высокой степени автокорреляции в пространственных данных. При решении задач пространственного извлечения знаний необходимо рассчитывать эту величину и внедрять ее в классические методы *data mining*. Задача пространственного извлечения знаний состоит в автоматизации обнаружения зависимостей между объектами или классами объектов и предоставлении полученных результатов для дальнейшего исследования.

В основу технологий классического и пространственного извлечения знаний положена концепция шаблонов (*patterns*), представляющих собой закономерности. В результате обнаружения этих скрытых от невооруженного глаза закономерностей решаются задачи извлечения знаний. Различным типам закономерностей, которые могут быть выражены в форме, понятной человеку, соответствуют определенные задачи. Наиболее часто решаемыми задачами в пространственном извлечении знаний являются задачи классификации, кластеризации и ассоциации.

**Задача классификации** — формализованная задача, в которой имеется множество объектов (ситуаций), разделённых некоторым образом на классы. Задано конечное множество объектов, для которых известно, к каким классам они относятся. Это множество называется выборкой. Классовая принадлежность остальных объектов неизвестна. Требуется построить алгоритм, способный классифицировать произвольный объект из исходного множества. Другими словами, необходимо оценить значение одного атрибута отношения на основе других атрибутов этого отношения. Частными случаями задачи классификации являются задача определения местоположения тех или иных объектов, например расположения станций технического обслуживания или магазинов в пределах города, и задача тематической классификации, которая тесно связана с анализом пространства, так как в большинстве ситуаций соседние объекты принадлежат к одному и тому же классу.

**Задача кластеризации** — задача разбиения заданной выборки объектов (ситуаций) на непересекающиеся подмножества, называемые кластерами, так чтобы каждый кластер состоял из схожих объектов, а объекты разных кластеров существенно отличались. Другими словами, с помощью алгоритмов классификации можно из огромного числа разнородных географических объектов выделить группы, или кластеры, которые формируются на основе критерия «сходства», используемого для определения связей между значениями характеристик объектов, и представить полученные результаты в качестве картографической основы для веб-ориентированной ГИС.

**Задача ассоциации** — установление взаимодействия между атрибутами. Быстрый поиск шаблонов в больших массивах данных — необходимое условие при формировании пространственной информации для ГИС. Один из самых простых и качественных приемов при этом — поиск закономерности между связанными событиями в наборе данных и выявление правил ассоциации. При решении задачи ассоциации устанавливается причинно-следственная связь между значениями характеристик объектов. Отличие задачи ассоциации от двух предыдущих задач извлечения знаний состоит в том, что поиск закономерностей осуществляется не на основе свойств анализируемого объекта, а между несколькими событиями, которые происходят одновременно.

Далее подготовка данных для веб-ориентированной ГИС состоит в определении способа организации хранения информации, полученной в

результате проведенного анализа. Существует множество подходов к хранению пространственных данных, но самым удобным, наиболее защищенным и доступным, по мнению авторов, является единое хранилище пространственных данных [20].

Остальные сервисы пространственного анализа выступают в качестве определенных прикладных функций веб-ориентированной ГИС, причем данные сервисы могут решать как общие задачи для разных ГИС (базовые сервисы анализа), так и узкоспециализированные задачи для каждой ГИС в отдельности (специализированные сервисы анализа).

### **Базовые сервисы анализа**

К общим сервисам анализа, используемым в ГИС, относятся:

- сервисы сетевого анализа;
- сервис тематического картографирования;
- сервис построения пространственных запросов.

**Сервисы сетевого анализа.** Сетевой анализ позволяет пользователю решать различные задачи на пространственных сетях связанных линейных объектов (реках, дорогах, трубопроводах, линиях электропередач и т. п.) в зависимости от области применения ГИС. В классическом представлении сеть считается набранной из линий, которые могут иметь не более двух общих точек с другими линиями — точки начала и конца. Математически сети описываются теорией графов, а решение многих сетевых задач осуществляется с помощью методов линейного программирования. Как правило, решением задач сетевого анализа является определение ближайшего, наиболее выгодного, кратчайшего пути, определение уровней нагрузки на сеть, определение зон влияния на объекты сети других объектов.

Основной целью реализации **сервиса тематического картографирования** является выделение классов (тем) пространственно распределенных объектов, имеющих схожие значения параметров, по которым и проводится тематическая классификация объектов. При построении тем пользуются методами классификации и кластеризации. Если число тем известно, то необходимо только определить принадлежность того или иного географического объекта определенному классу, т. е. воспользоваться методами классификации. В случае когда число классов неизвестно, применяя методы кластеризации, выявляют эти классы и соответствующие им объекты. Результатом работы данного сервиса является построение тематически окрашенной карты.

Основанием для реализации *сервиса построения пространственных запросов* является присущее пространственным объектам свойство находиться в топологическом отношении друг с другом и описываться теми или иными геометрическими характеристиками, такими как длина, периметр, площадь.

Топологические операции включают установление отношений смежности, связности между объектами. При проведении анализа важной информацией является взаимное расположение объектов относительно друг друга. СУБД, являющиеся составной частью ГИС, поддерживают составление запросов на выборку объектов по определенным условиям. При создании таких запросов могут учитываться не только операторы отношений (равно, больше, меньше и др.), логические операторы (и, или, не), арифметические операторы (сложение, вычитание и др.), но и пространственные операторы (соприкасаются, пересекаются, не пересекаются, содержат, быть внутри, покрывать, покрываться, равны). Пространственные операторы отображают межобъектные топологические отношения между объектами карты [29].

Геометрический анализ включает измерение пространственных характеристик, объектов. Данный тип анализа наиболее широко распространен, так как не требует сложных вычислений и включает в себя функции расчета площадей, длин, периметров объектов, измерения расстояния на карте.

Реализация сервиса подразумевает поиск объектов пространственной базы данных по различным критериям. Критериями в данном случае служат топологические и геометрические свойства объектов. Основным инструментом для поиска объектов является конструктор пространственных запросов, при помощи которого происходит определение условий поиска объектов. Результатом запроса является выборка объектов, представленная в виде таблицы, отдельного слоя на электронной карте либо выделение объектов, удовлетворяющих условиям поиска.

### **Специализированные сервисы анализа**

Набор данных сервисов определяется в зависимости от задач, решаемых с помощью самой ГИС. Например, если ГИС применяется в области здравоохранения, то важно знать и уметь выявлять области распространения того или иного заболевания в зависимости от различных факторов (социально-экономических, природных и др.). При использовании ГИС для управления инженерными сетями с применением методов

пространственного анализа можно рассчитать и определить местоположение водопроводных колодцев, затапливаемых талой водой весной, и т. д. В зависимости от способа решаемой задачи узкоспециализированные сервисы могут использовать методы и алгоритмы как *spatial data mining*, так и базовых сервисов.

### **Сервисы представления результатов анализа**

Результаты выполнения функций пространственного анализа можно представлять в различном формате в зависимости от типа выполняемого анализа, но наиболее наглядным отображением являются электронные карты. Операции зонирования могут быть основаны на формальных методах кластерного анализа в пространстве признаков и перенесении полученных результатов кластеризации в географическое пространство с последующим отображением на электронной карте в простом и наглядном виде. Результаты выполнения запросов к пространственной базе данных могут содержать десятки объектов, поэтому их лучше всего представить в табличном виде с возможностью отображения полученной выборки или ее части.

Предложенный набор сервисов для проведения пространственного анализа в среде веб-ориентированной ГИС включен в архитектуру ГИС для управления инженерными сетями крупных промышленных предприятий и муниципальных образований.

# Глава 5

## ВЕБ-ОРИЕНТИРОВАННАЯ ГЕОИНФОРМАЦИОННАЯ СИСТЕМА WGS3

---

### 5.1. Администрирование. Управление проектом

Создание веб-ориентированной геоинформационной системы WGS3 (Web-Gis-Server) обусловлено необходимостью повышения эффективности применения геоинформационных технологий для управления инженерной инфраструктурой промышленных предприятий и муниципальных образований. Основными задачами, решаемыми геоинформационной системой WGS3, являются:

- максимально быстрое получение с помощью электронного сетевого оборудования точной информации об объектах инженерных сетей для выполнения смежных работ;
- многократное распараллеливание одновременной работы с готовыми и точными планами инженерных сетей, позволяющее ускорить выполнение проектно-конструкторских и ремонтно-строительных работ вследствие высокого качества информации;
- максимальная скорость поиска информации о сосредоточенных и распределенных объектах;
- удобство привязки информации к графическим объектам.

Преимущества использования данной системы:

- предоставление оперативного доступа к электронным данным планов по инженерным сетям в среде корпоративной сети предприятия;
- использование различных форм доступа к коммерческим и служебным данным;
- доступность для пользователей инженерно-технических подразделений, имеющих навыки работы с компьютером и в сети Интернет;
- снижение затрат за счет использования распределенной ГИС и применения архитектуры «тонкий клиент».

Веб-ориентированная геоинформационная система WGS3 представляет собой веб-приложение, работающее на базе сетевой ГИС Autodesk MapGuide.

В состав системы WGS3 включены три модуля:

- 1) подсистема «Управление проектом»;
- 2) подсистема «Управление данными»;
- 3) пользовательские функции системы.

**Подсистема «Управление проектом»** предоставляет администратору системы следующие возможности:

- управление пользователями (рис. 5.1) и ролями (рис. 5.2);
- определение параметров доступа к объектам проекта: стандартным и системным функциям, панелям, слоям с данными и подложкам;
- назначение карты, выводимой пользователям системы по умолчанию для проекта и для окна общего вида.

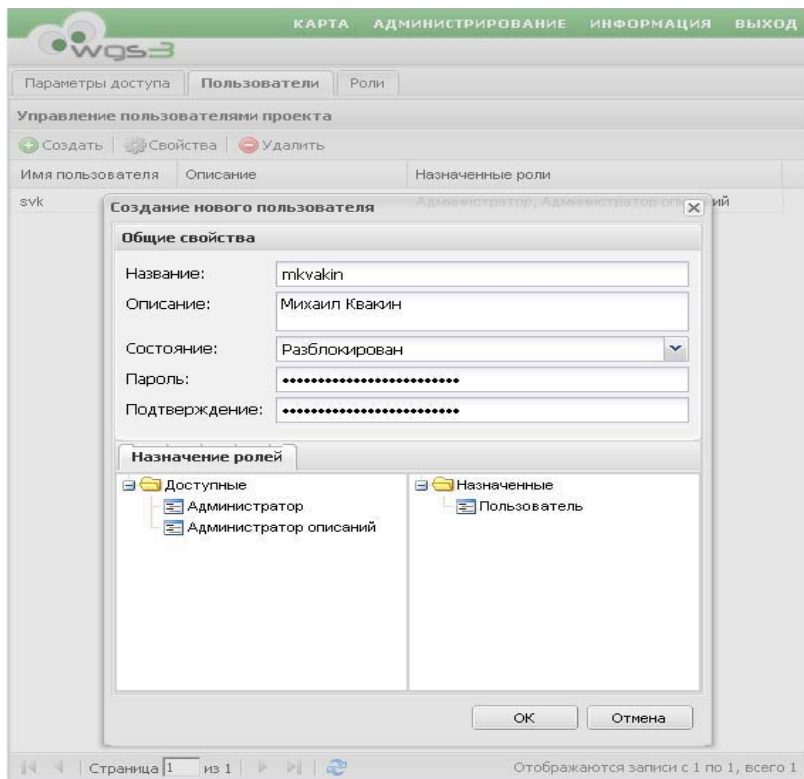


Рис. 5.1. Экранная форма «Создание нового пользователя»

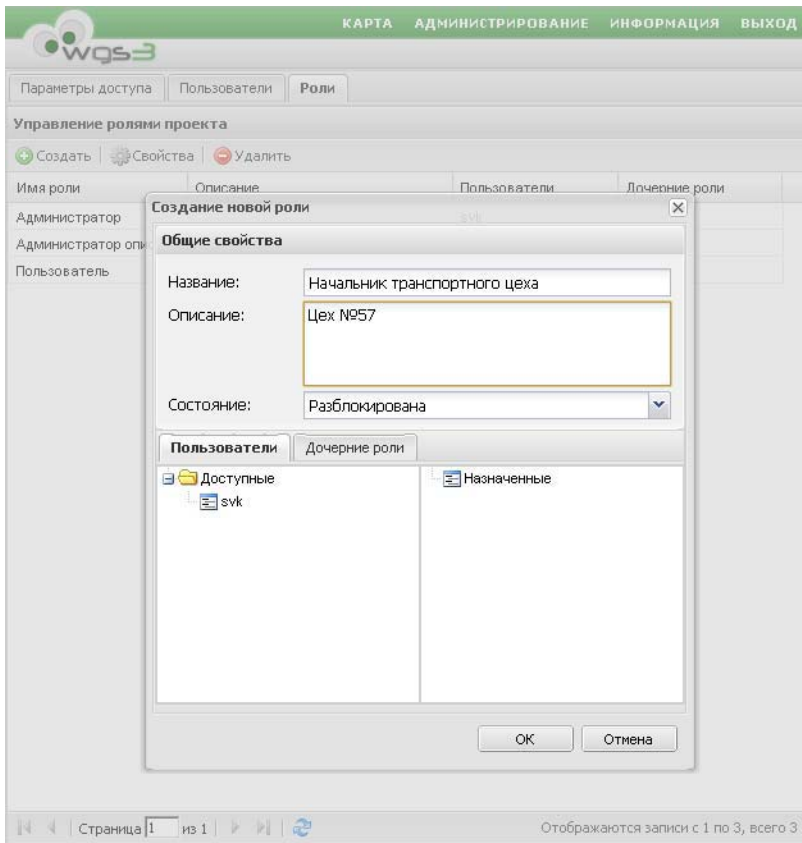


Рис. 5.2. Экранная форма «Создание новой роли»

Назначение параметров доступа ролей к объектам проекта осуществляется в двух режимах:

1) *режим назначения прав через объекты*. Рекомендуется использовать в случае, если необходимо назначить нескольким ролям права на один объект (рис. 5.3);

2) *режим назначения прав через роли*. Рекомендуется использовать в случае, если необходимо назначить одной роли права на несколько объектов (рис. 5.4).



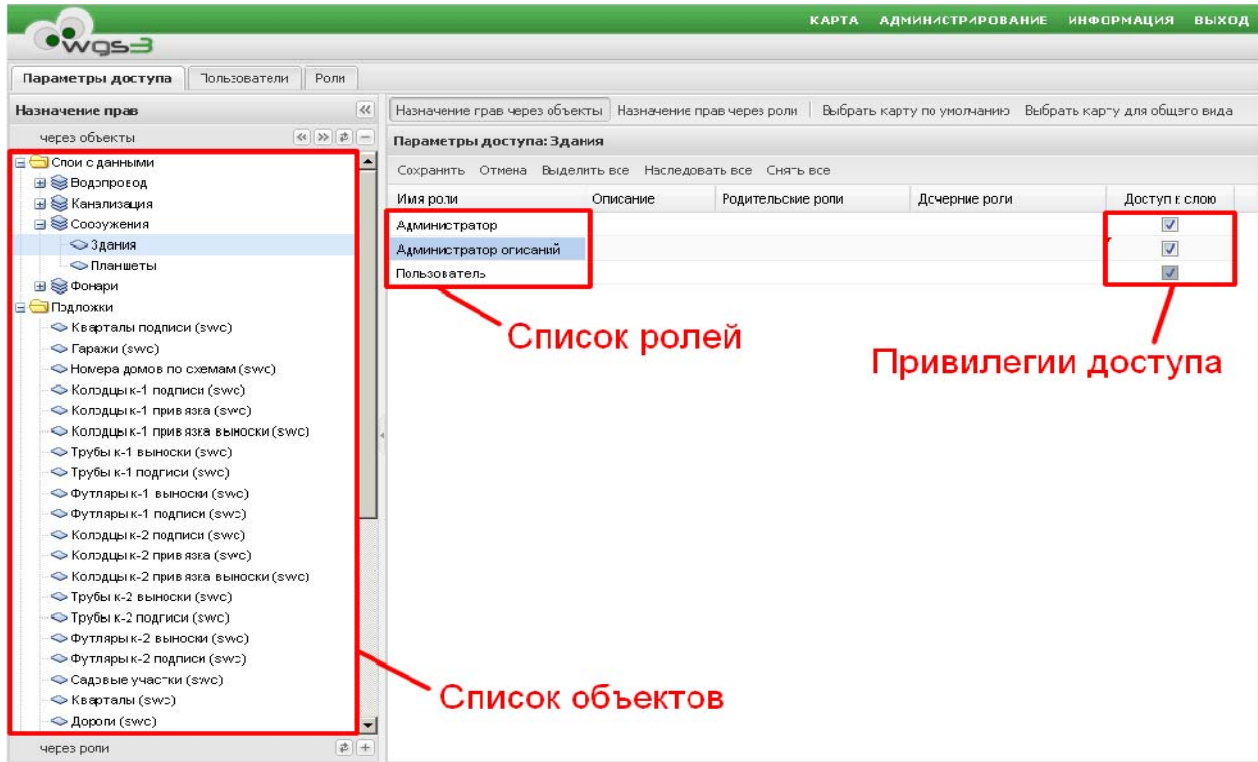


Рис. 5.3. Назначение прав через объекты

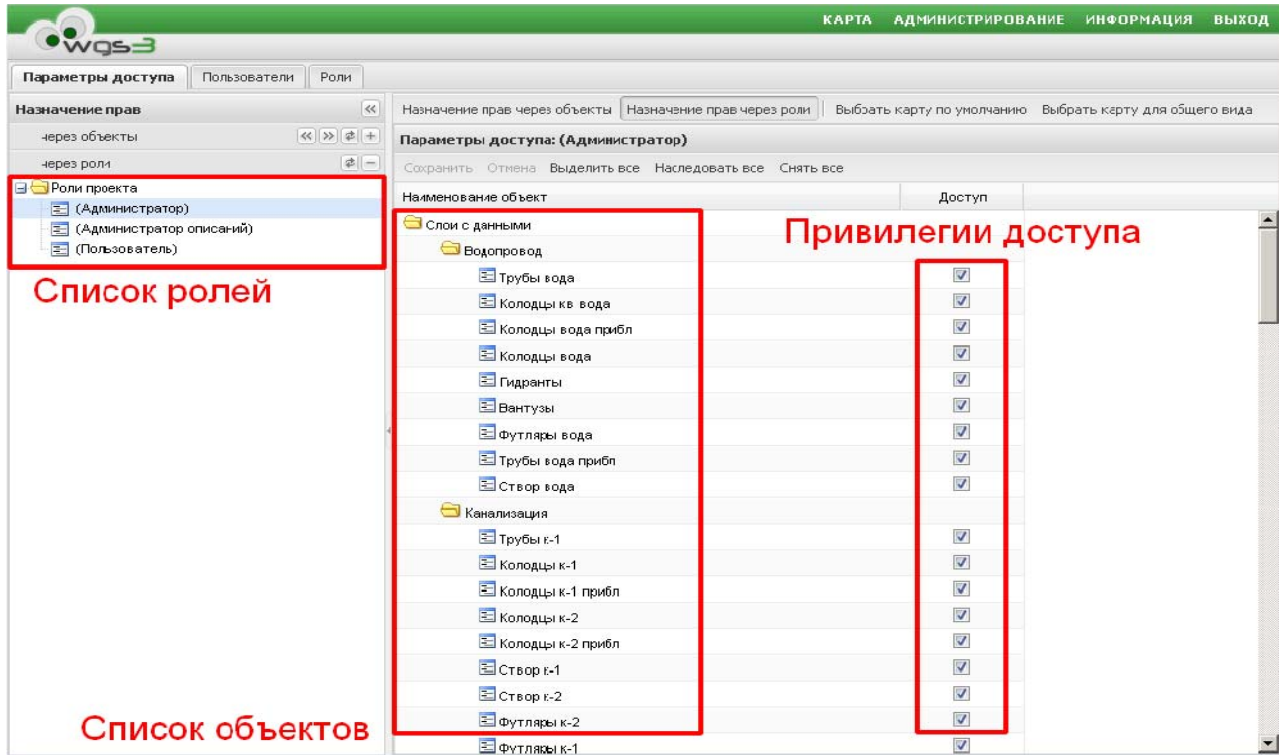


Рис. 5.4. Назначение прав через роли

Назначение прав осуществляется присвоением привилегий доступа:

- *назначить привилегию* (например, «доступ к слою»);
- *снять привилегию*;
- *наследовать параметр от родительской роли*.

Проект может содержать несколько видов представлений пространственной информации (карт). Настройка подключения конкретной карты показана на рис. 5.5 (выбор карты по умолчанию) и рис. 5.6 (выбор карты для окна общего вида).

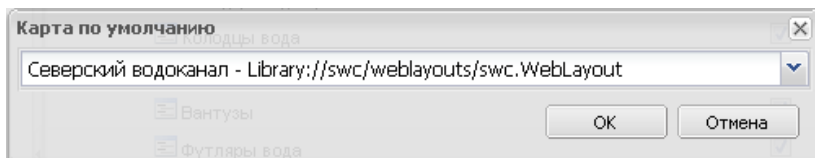


Рис. 5.5. Экранная форма «Выбор карты по умолчанию»

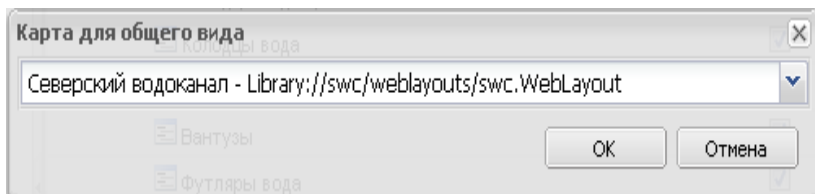


Рис. 5.6. Экранная форма «Выбор карты для окна общего вида»

## 5.2. Администрирование. Управление данными

В геоинформационной системе WGS3 выделяется пять типов данных:

- 1) атрибутивные описания;
- 2) слои;
- 3) группы слоев;
- 4) характеристики;
- 5) группы характеристик.

**Атрибутивное описание** представляет собой описательную структуру объектов реального мира, состоящую из множества свойств (характеристик). Каждое описание может иметь простую либо сложную

иерархическую структуру. При использовании иерархической структуры каждое дочернее (вложенное) описание вступает в связь типа «один-к-одному» или «один-ко-многим». Такая связь, в свою очередь, также может иметь свойства (характеристики). Карточка дочернего описания в иерархической структуре именуется как *вложенная*.

Сущность связи «один-к-одному» состоит в том, что экземпляр (карточка) родительского атрибутивного описания может иметь только один экземпляр (карточку) дочернего атрибутивного описания. Например, карточка типа «Помещение» может иметь одну дочернюю карточку типа «Пол». Связь «один-ко-многим» показывает, что экземпляр (карточка) родительского атрибутивного описания может иметь множество экземпляров (карточек) дочернего атрибутивного описания. Например, карточка типа «Помещение» может иметь несколько дочерних карточек типа «Компьютер». Такая связь может называться «Компьютеризация» и иметь свойство «Общая стоимость».

**Слой** представляет собой класс объектов реального мира, объединенных общей тематикой. Слой может быть связан с одним или несколькими атрибутивными описаниями, в результате чего каждый объект слоя будет связан с карточками соответствующих атрибутивных описаний (будет описываться совокупностью их характеристик). В случае когда слой связан с атрибутивными описаниями, одно из них должно быть назначено в качестве *основного*, а остальные — *дополнительных*.

Отличие основного атрибутивного описания от дополнительного состоит в следующем:

- карточка основного описания отображается первой при использовании функции «Карточка»;
- при использовании функции «Поиск» в режиме поиска объектов на карте в списке результатов отображаются карточки основного атрибутивного описания;
- выводимая подпись к объектам на карте выводится из карточки основного описания.

**Группы слоев** для удобства управления выделяются по тематическому признаку (например, группа «Водопровод», включающая слои «Вода пожарная» и «Вода техническая»).

**Характеристика** представляет собой описательную единицу объекта реального мира. Набор характеристик является атрибутивным описанием.

**Группы характеристик**, как и группы слоев, выделяются в целях удобства управления. Например, характеристики «Номер дома» и «Улица» удобно объединить в группу адресных характеристик.

Главное окно подсистемы управления данными разделено на две области: слева представлено дерево объектов, сгруппированное по типам; справа — рабочая область подсистемы, в которой каждый редактируемый или создаваемый объект представляется в виде формы, открываемой в отдельной вкладке (рис. 5.7).

**Подсистема «Управление данными»** предоставляет администратору следующие возможности:

- управление слоями (рис. 5.8) и группами слоев (рис. 5.9);
- управление атрибутивными описаниями (рис. 5.10);
- управление группами характеристик (рис. 5.11) и самими характеристиками (рис. 5.12);
- связывание атрибутивных описаний со слоями;
- управление атрибутивными описаниями (рис. 5.13) или выполнение поиска по этим описаниям;
- управление отвязанными, потерянными или неиспользуемыми атрибутивными описаниями;
- привязка атрибутивных описаний к объектам карты.

**Для привязки атрибутивного описания** в экранной форме привязки необходимо выбрать необходимое описание с учетом трех параметров (см. рис. 5.10):

1) **«назначить данное описание основным»**. После привязки данное описание будет назначено основным. Если до момента назначения основным было другое описание, то оно автоматически будет назначено как дополнительное;

2) **«привязать имеющиеся атрибутивные объекты к подходящим геометрическим»**. Если выбранное атрибутивное описание уже было ранее привязано к слою и в системе сохранились отвязанные карточки данного описания, то вместо создания пустых карточек для каждого объекта слоя будут восстановлены связи со старыми карточками. Если в слое появились новые объекты, то для них будут созданы новые пустые карточки;

3) **«удалить все атрибутивные объекты, не привязанные к геометрическим»**.

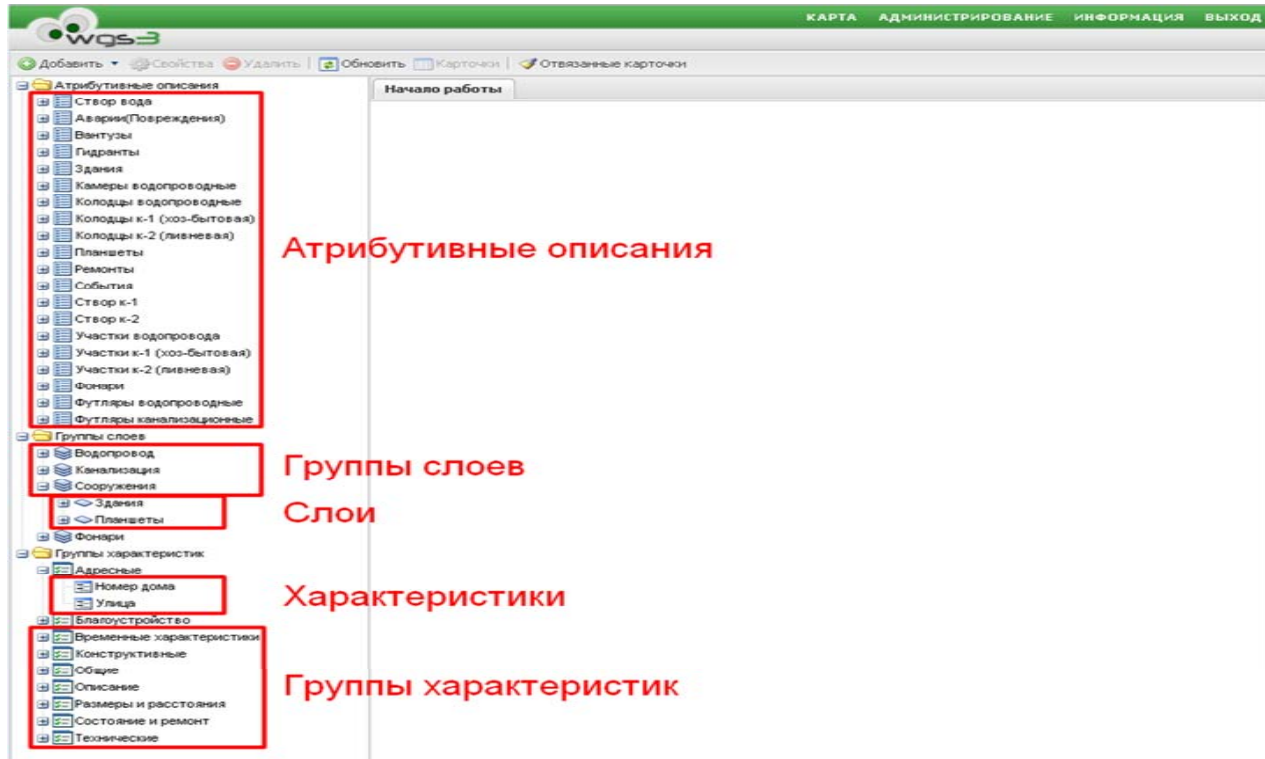


Рис. 5.7. Главное окно подсистемы управления данными

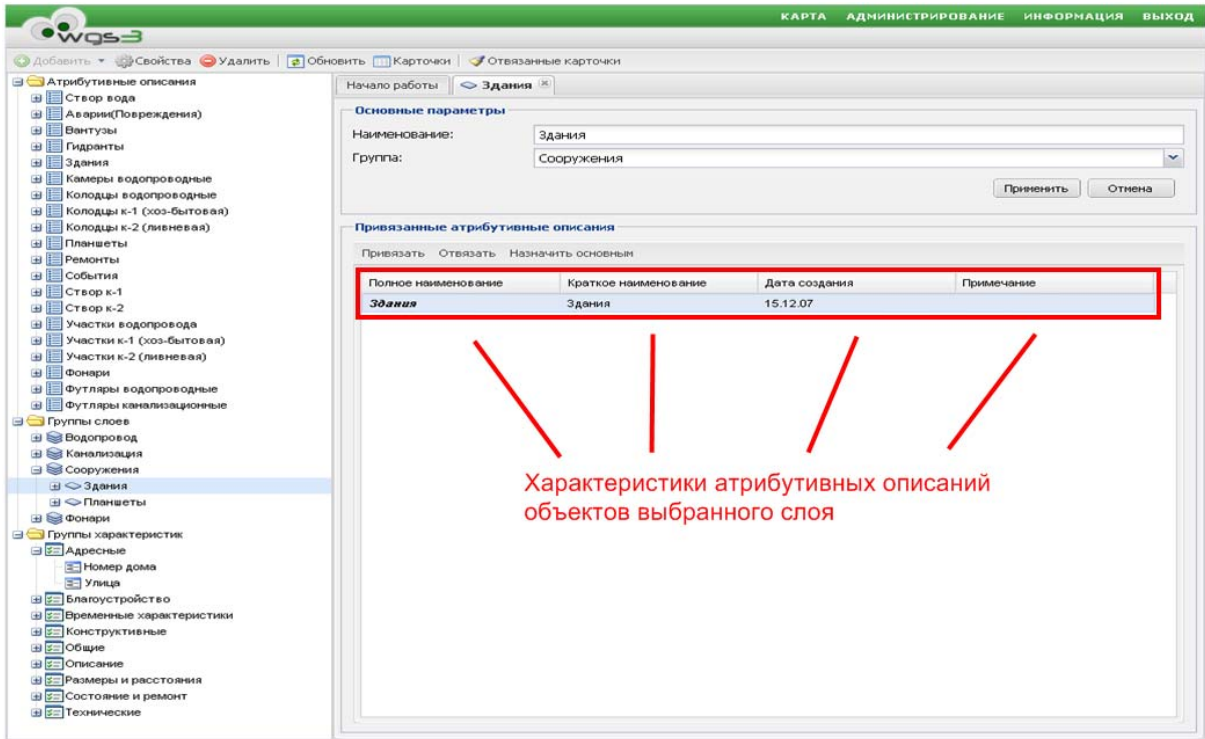


Рис. 5.8. Форма создания слоя

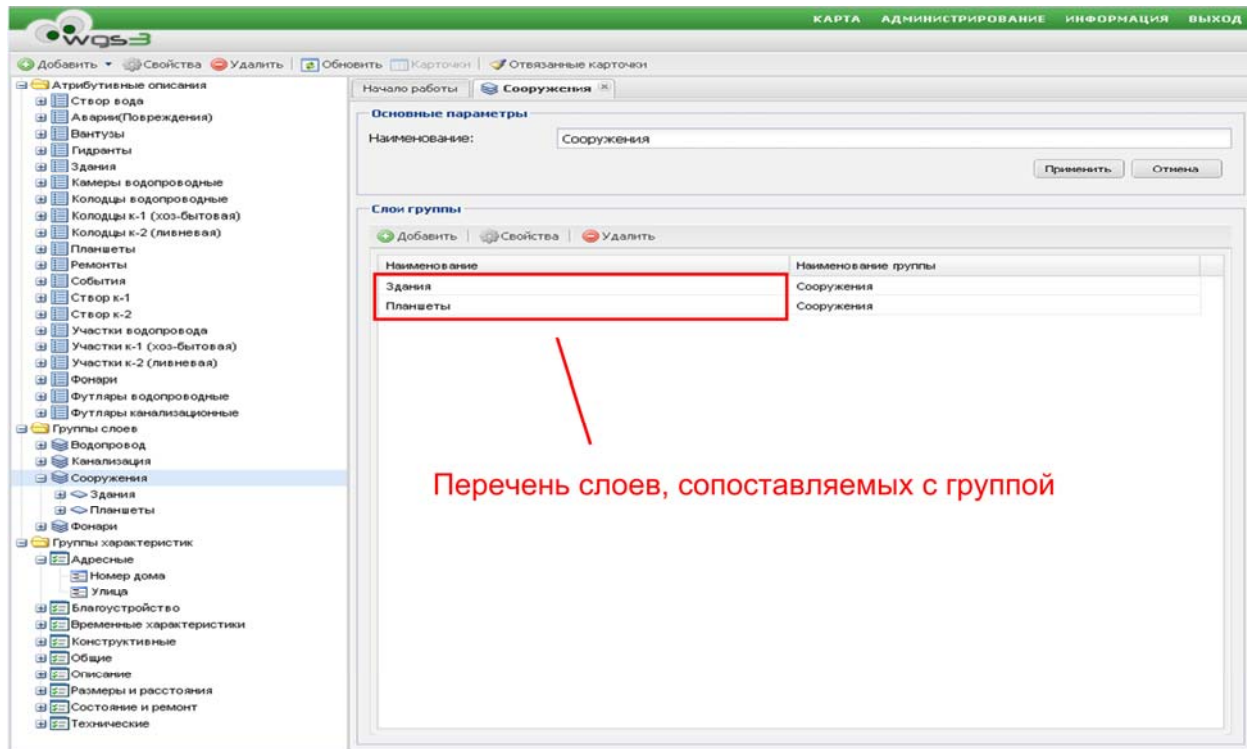


Рис. 5.9. Форма создания группы слоев



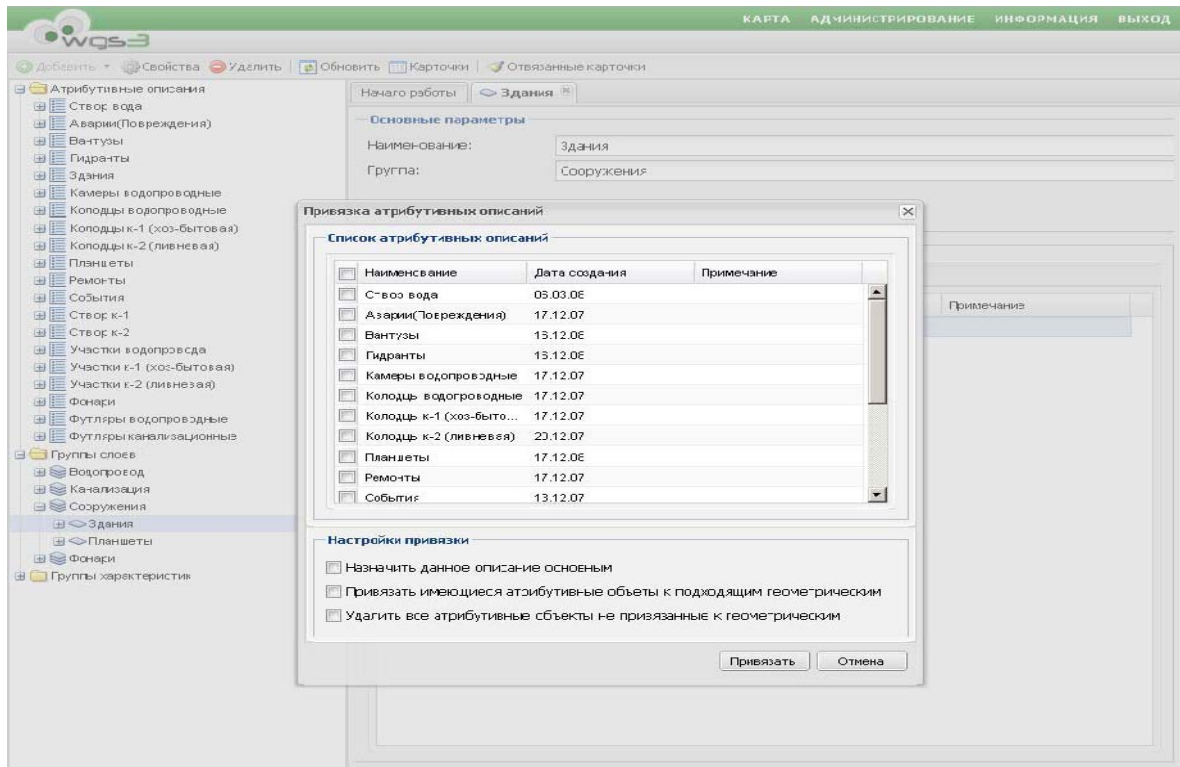


Рис. 5.10. Экранная форма «Привязка атрибутивного описания»

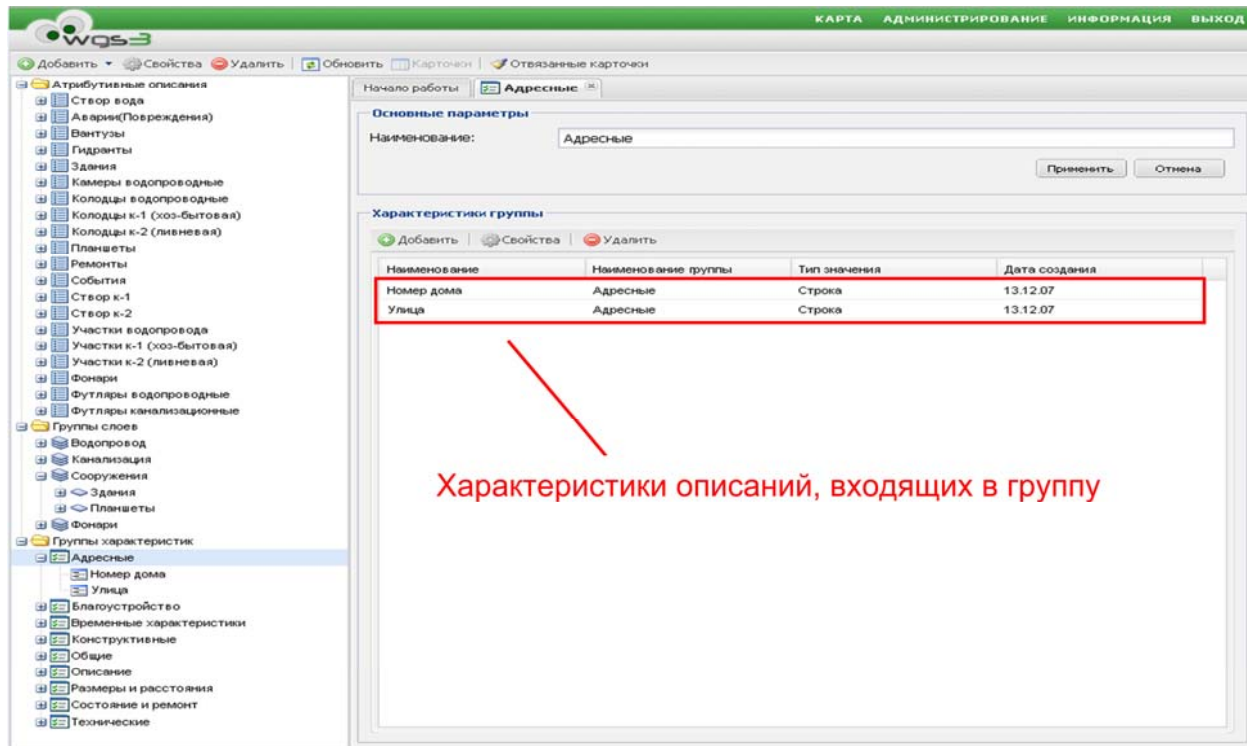


Рис. 5.11. Экранная форма «Создание группы характеристик»

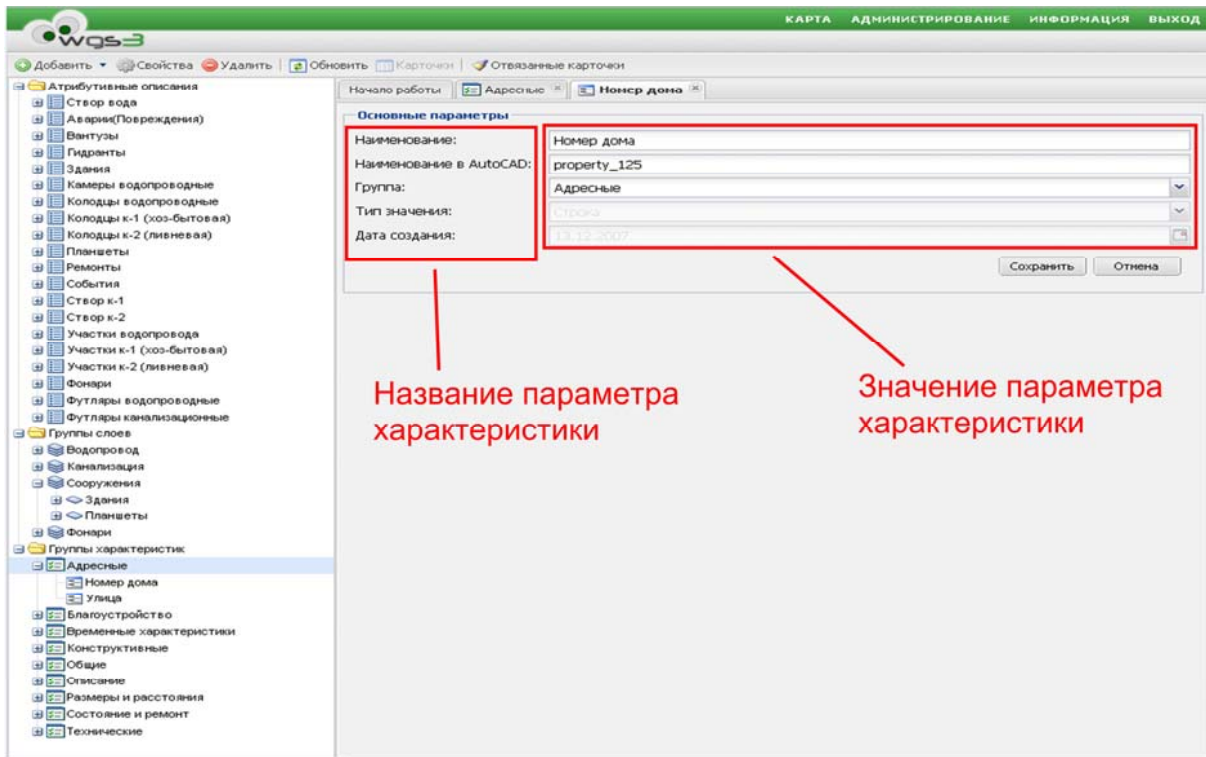


Рис. 5.12. Экранная форма «Создание характеристики»

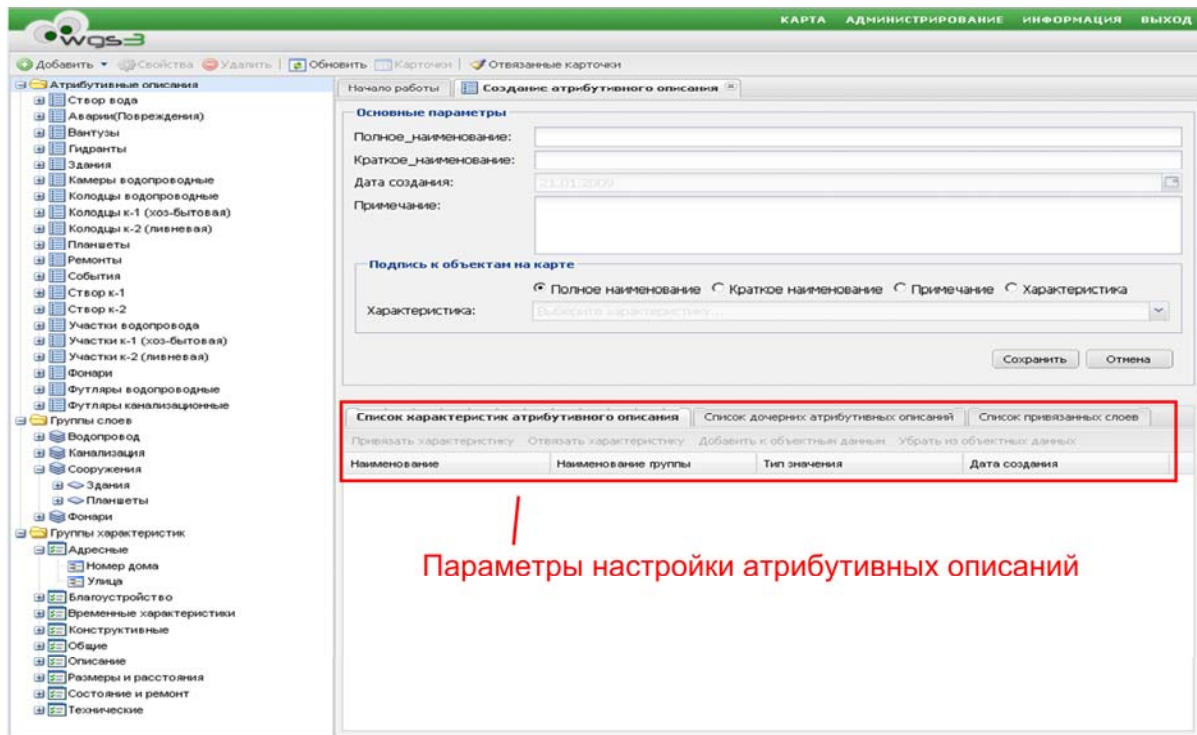


Рис. 5.13. Экранная форма «Создание атрибутивного описания»

Параметр «Удалить все атрибутивные объекты, не привязанные к геометрическим» зависит от параметра «Привязать имеющиеся атрибутивные объекты к подходящим геометрическим»:

– *параметр включен*. Если в системе сохранились отвязанные карточки данного описания, то те из них, связи с которыми не удалось восстановить, будут удалены;

– *параметр выключен*. Если в системе сохранились отвязанные карточки данного описания, то они будут удалены.

Назначение основного описания осуществляется выбором описания из списка связанных атрибутивных описаний и нажатием кнопки «Назначить основным». После назначения основного описания оно выделяется жирным стилем.

**При создании характеристики** автоматически создаются следующие параметры:

– наименование характеристики, используемое при отображении в карточке и форме поиска;

– наименование в AutoCAD (синоним), которое будет использоваться в таблице объектных данных при работе с AutoCAD Map (Autodesk Map). При загрузке объектов из среды AutoCAD Map в базу данных значение из одноименного поля будет загружено в соответствующую характеристику карточки. Длина наименования не должна превышать 15 символов, допустимыми символами являются латинские и русские буквы, символ «\_» и цифры;

– группа характеристик, в которую помещается данная характеристика;

– тип значения характеристики, выбираемый из списка:

- «Дата-Время»;
- «Список значений»;
- «Ссылка на объект»;
- «Строка»;
- «Текст»;
- «Число вещественное»;
- «Число целое».

**В случае создания нового атрибутивного описания** у данного описания автоматически создаются следующие параметры (см. рис. 5.13):

– полное наименование, выводимое в карточке и в списке атрибутивных описаний;

- краткое (сокращенное) наименование атрибутивного описания, не являющееся обязательным при заполнении;
- примечание — дополнительная информация об атрибутивном описании;
- подпись к объектам на карте — характеристика, значение которой будет выводиться в виде подписи либо подсказки при отображении на карте.

После сохранения атрибутивного описания оно становится доступным для редактирования. В нижней части формы доступны три раздела:

- 1) список характеристик атрибутивного описания;
- 2) список дочерних атрибутивных описаний;
- 3) список привязанных слоёв.

**Список характеристик атрибутивного описания.** Для привязки характеристики к атрибутивному описанию необходимо нажать кнопку «Привязать характеристику». В открывшейся форме привязки отметить галочкой нужные характеристики (рис. 5.14).

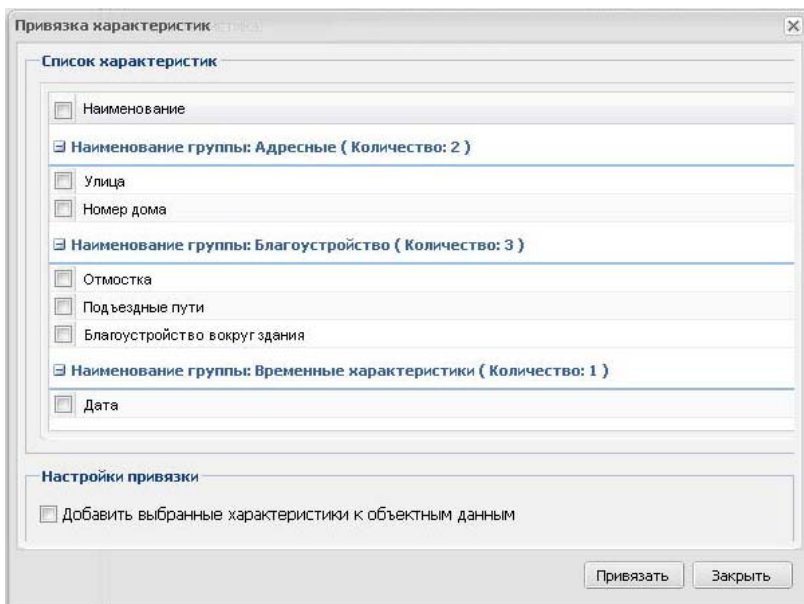


Рис. 5.14. Экранная форма «Привязка характеристик к атрибутивному описанию»

Если характеристики нужно добавить в таблицу объектных данных, то следует установить параметр «Добавить выбранные характеристики к объектным данным». Для «отвязки» характеристики необходимо выбрать характеристику из списка и нажать кнопку «Отвязать характеристику». С помощью кнопок «Добавить к объектным данным» и «Убрать из объектных данных» производится добавление или удаление уже привязанных к описанию характеристик из таблицы объектных данных.

**Список дочерних атрибутивных описаний.** Для создания, редактирования и удаления связей с дочерними атрибутивными описаниями используются кнопки «Добавить», «Свойства» и «Удалить». Для создания связи необходимо нажать кнопку «Добавить». В открывшейся форме (рис. 5.15) указываются наименование, тип связи и атрибутивные описания, участвующие в связи. После этого нажать на кнопку «Сохранить».

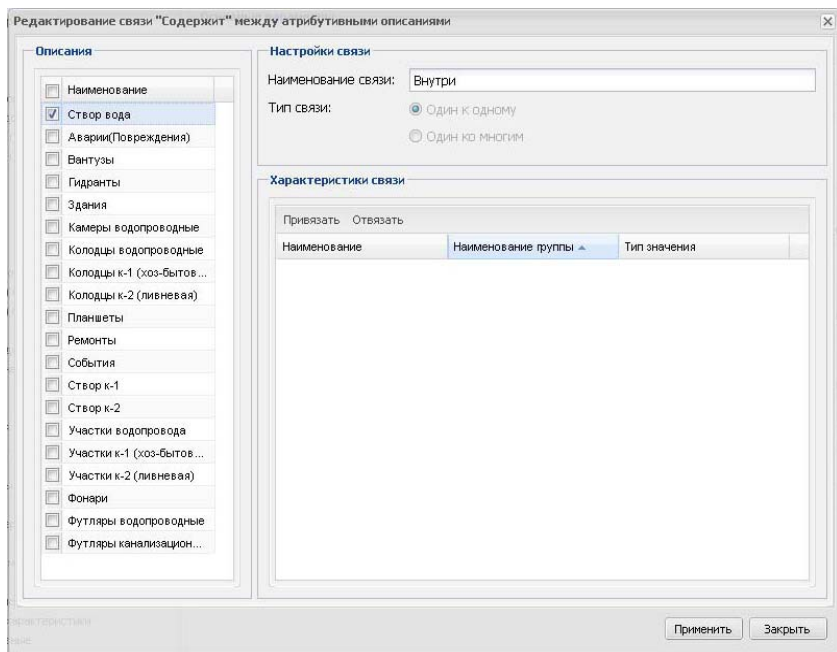


Рис. 5.15. Экранная форма «Создание связи с дочерним атрибутивным описанием»

После сохранения отредактированной связи становится доступной возможность назначения характеристик связи (рис. 5.16). Для этого необходимо использовать кнопки «Привязать» и «Отвязать».

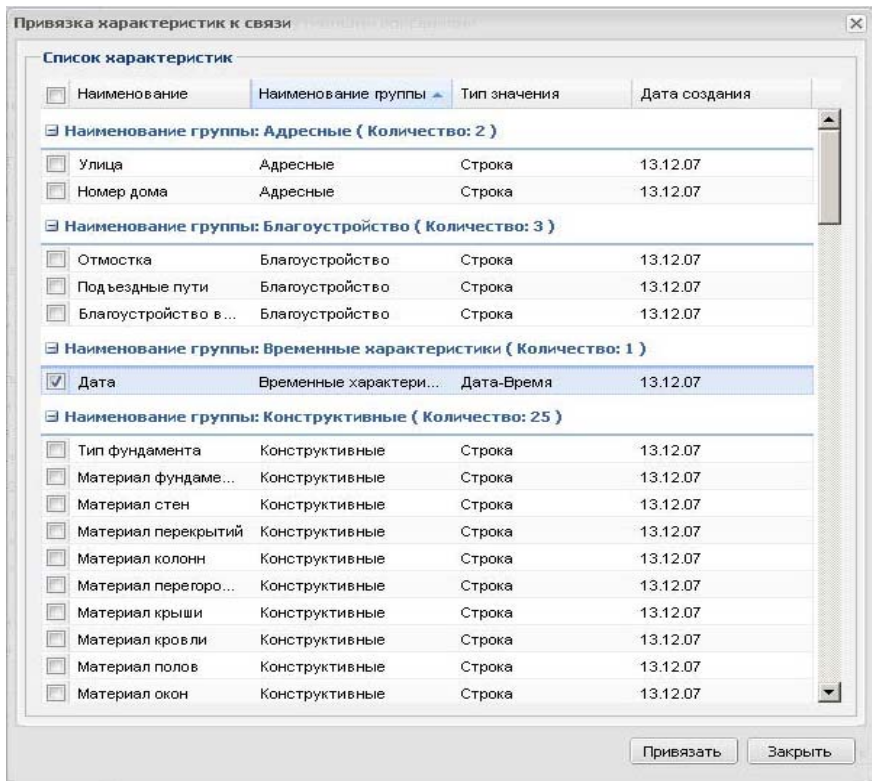


Рис. 5.16. Экранная форма «Назначение характеристик связи»

**Список привязанных слоев.** Для привязки атрибутивного описания к слоям необходимо в списке выбрать слой и нажать кнопку «Привязать». В форме параметров привязки необходимо установить галочки в соответствии с правилами.

Помимо просмотра карточек с атрибутивным описанием в подсистеме управления данными предусмотрена возможность **управления «потерянными» карточками** следующими способами:



- вручную при работе с модулем привязки карточек — при отвязке дополнительной или вложенной карточки;
- при отвязке атрибутивного описания от слоя;
- при удалении объекта с карты из среды AutoCAD Map.

Для просмотра отвязанных карточек необходимо нажать кнопку «Отвязанные карточки» на панели управления подсистемы управления данными (см. рис. 5.7). При работе с отвязанными карточками можно использовать три типа связей карточек:

- 1) карточки типа связи «Слой – Тип», оставшиеся в результате отвязки атрибутивного описания от слоя;
- 2) карточки типа связи «Объект – Карточка», оставшиеся в результате отвязки дополнительной карточки от геометрического объекта (через модуль «Привязка карточек»);
- 3) карточки типа связи «Карточка – Карточка», оставшиеся в результате отвязки вложенной карточки от родительской (через модуль «Привязка карточек»).

Отвязанные карточки можно привязать нажатием иконки привязки в списке карточек либо удалить, выделив карточки галочкой в списке и нажав кнопку «Удалить».

### 5.3. Пользовательские функции системы

Пользовательские функции системы представлены в модуле «Карта». **Общий вид окна** данного модуля показан на рис. 5.17. В левой части окна расположена *панель слоев*, позволяющая управлять тематическими слоями карты (все объекты на карте в геоинформационной системе группируются в наборы слоев определенной тематики). Слои могут быть иерархически объединены в общие группы. В панели слоев есть возможность включать или отключать отображение слоев, а также сворачивать или разворачивать списки групп слоев.

Вверху окна присутствует *панель инструментов*, содержащая программные кнопки, позволяющие производить вызов функций, присущих классическим ГИС-вьюерам (изменение вида и масштаба изображения, измерение расстояния и т. п.), а также и дополнительных специфических функций WGS3 (получение описания по объектам инженерных сетей, формирование топологических и геометрических запросов, подключение карты Google Maps и т. п.).

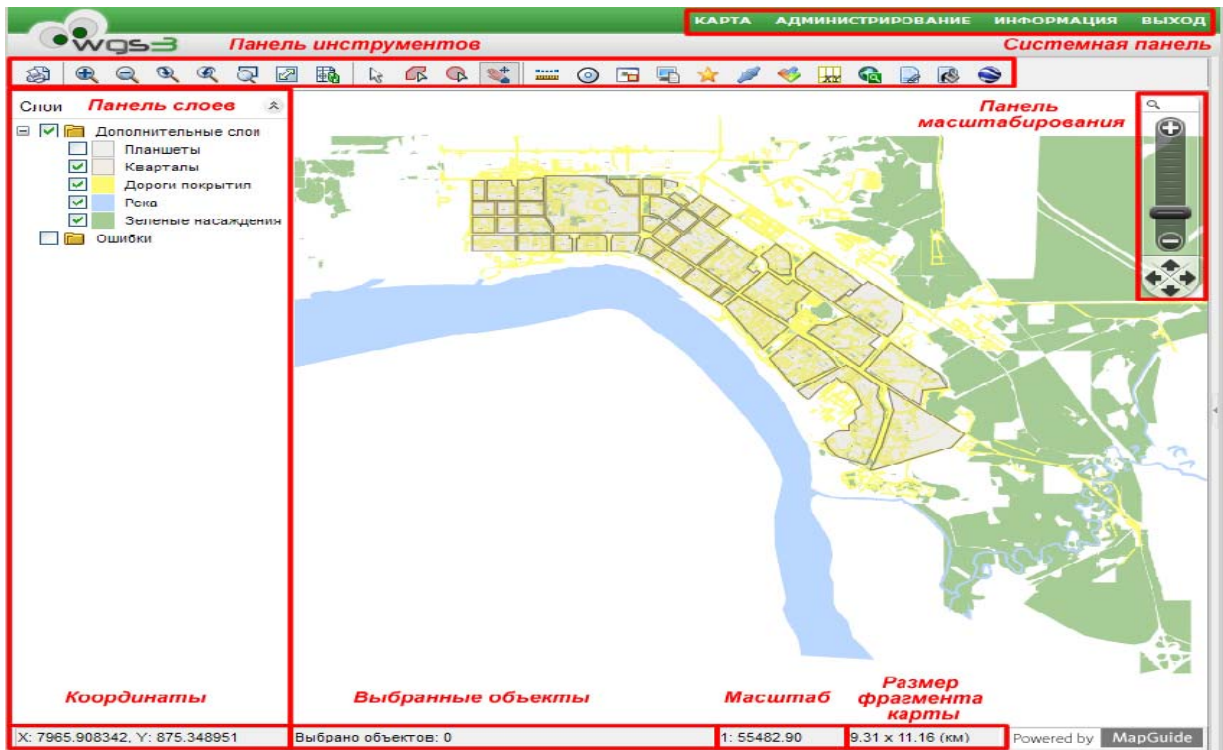


Рис. 5.17. Общий вид модуля «Карта»

Внизу окна присутствует *строка статуса*, которая может содержать в зависимости от прав пользователя координаты курсора, количество выбранных объектов, масштаб, размер фрагмента карты.

Пользователю доступна возможность управления масштабом через *панель масштабирования*, расположенную в правой части окна.

Начать работу с электронной картой пользователь может через **окно избранных видов**. Избранным видом называется часто используемая область карты (рис. 5.18). В системе WGS3 предусмотрено использование девяти избранных видов карты. При щелчке по объекту «Вид у пользователя» откроется окно, представленное на рис. 5.17, с необходимым позиционированием по координатам. Избранный вид сохраняет информацию о состоянии слоев карты (включен/отключен).

Объекты на карте можно выбирать тремя способами:

- 1) **выбор объекта прямоугольником** (рис. 5.19);
- 2) **выбор объекта полигоном** (рис. 5.20);
- 3) **выбор объекта окружностью** (рис. 5.21).

Механизм анализа графической информации в системе WGS3 также работает с использованием трех функций:

- 1) **измерение расстояния** (рис. 5.22);
- 2) **построение буферной зоны** (рис. 5.23);
- 3) **геометрический анализ объектов** (рис. 5.24).

В системе WGS3 геометрическим объектам системы соответствуют атрибутивные описания. К функциям управления атрибутивными описаниями относятся: *электронный документ, привязка объектов, поиск объектов, атрибутивное описание, множественное заполнение*.

К графическим объектам карты можно прикреплять различные электронные документы. Поддерживаемые форматы файлов: jpe, jpg, jpeg, png, gif, dwf, pdf, tif, tiff, avi, mpg, wmf.

На рис. 5.25 представлено окно, в котором можно осуществить **поиск геометрических объектов** и карточек по различным критериям.

В верхней части формы поиска находятся два выпадающих списка слоев и атрибутивных описаний. Эти данные можно не учитывать, и тогда поиск будет осуществляться по всем слоям и типам. В центральной части формы находится выпадающий список характеристик, отфильтрованных по выбранным типам. Простой поиск позволяет создавать запросы по значениям характеристик, связанным логическим «И» (характеристика1 содержит «значение1» И характеристика2 содержит «значение2»).

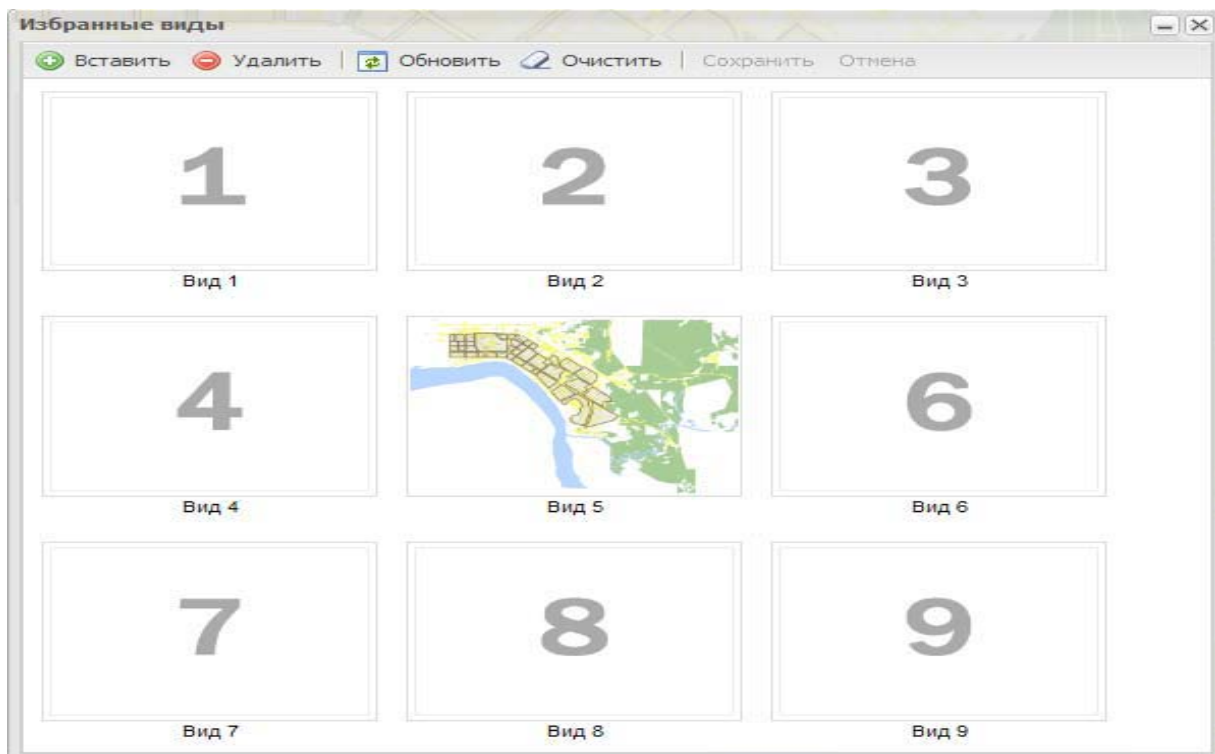


Рис. 5.18. Экранная форма «Управление избранными видами»

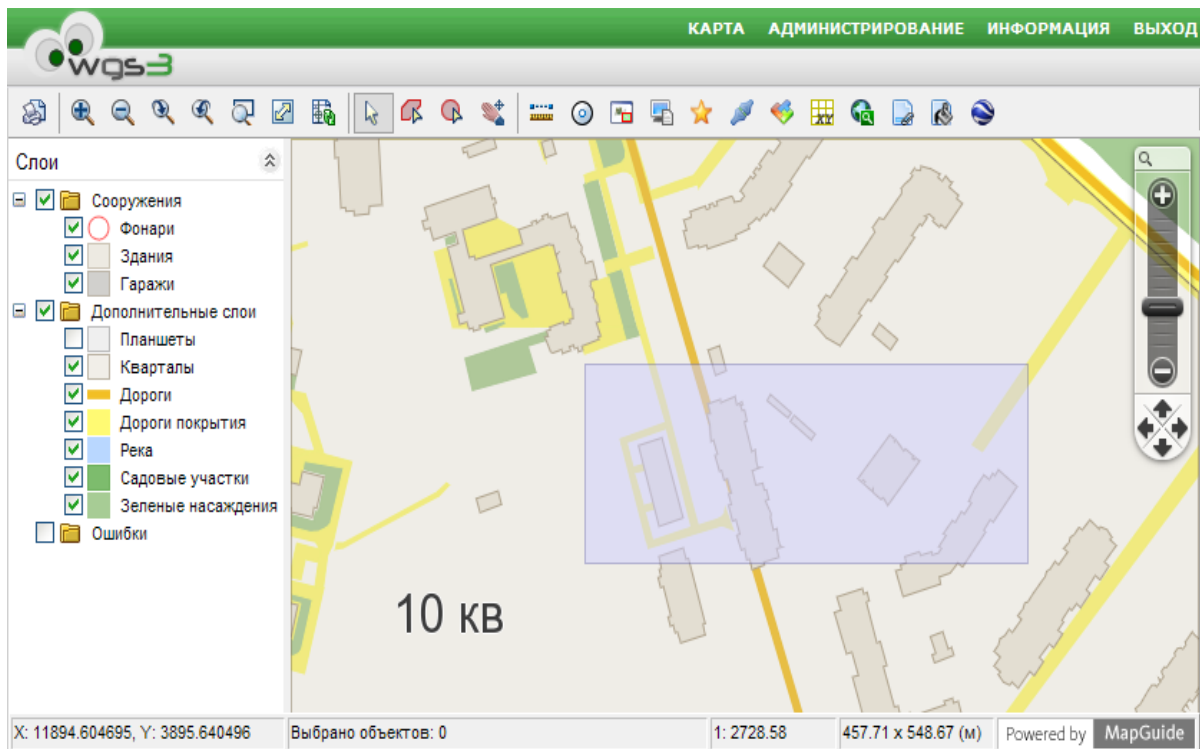


Рис. 5.19. Выбор объектов прямоугольником

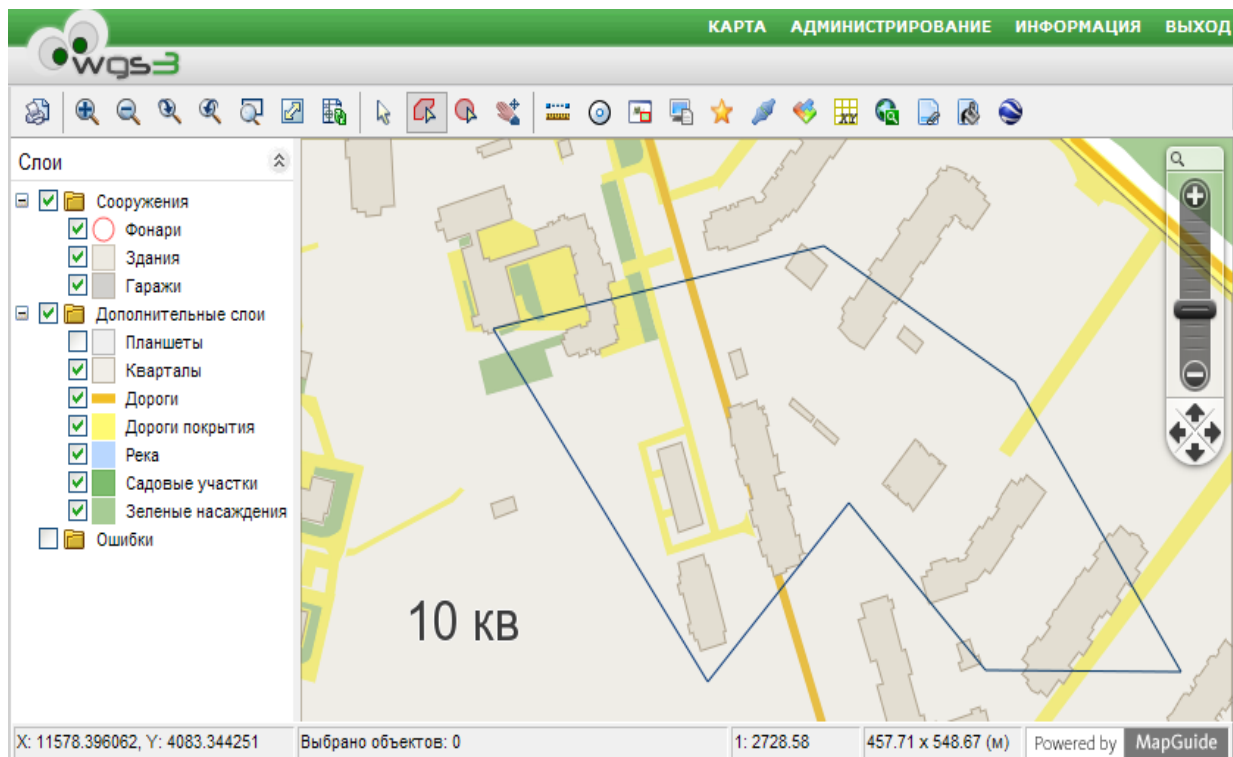


Рис. 5.20. Выбор объектов полигоном

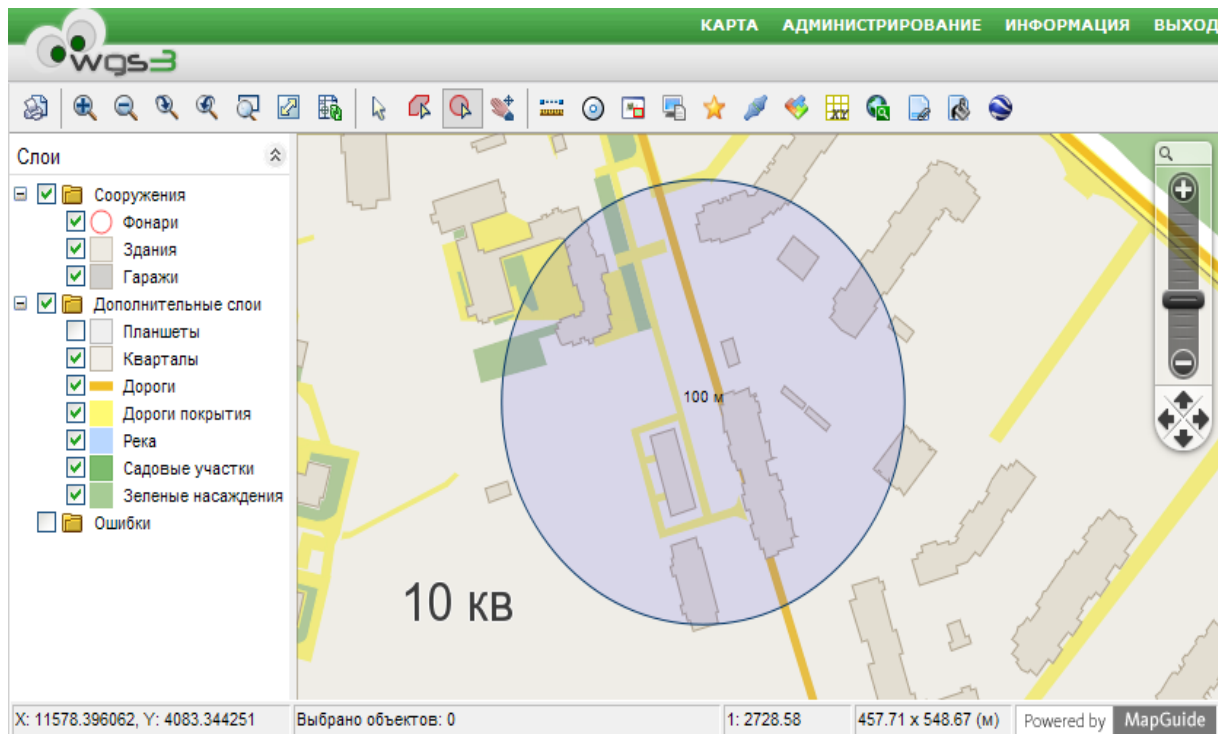


Рис. 5.21. Выбор объектов окружностью

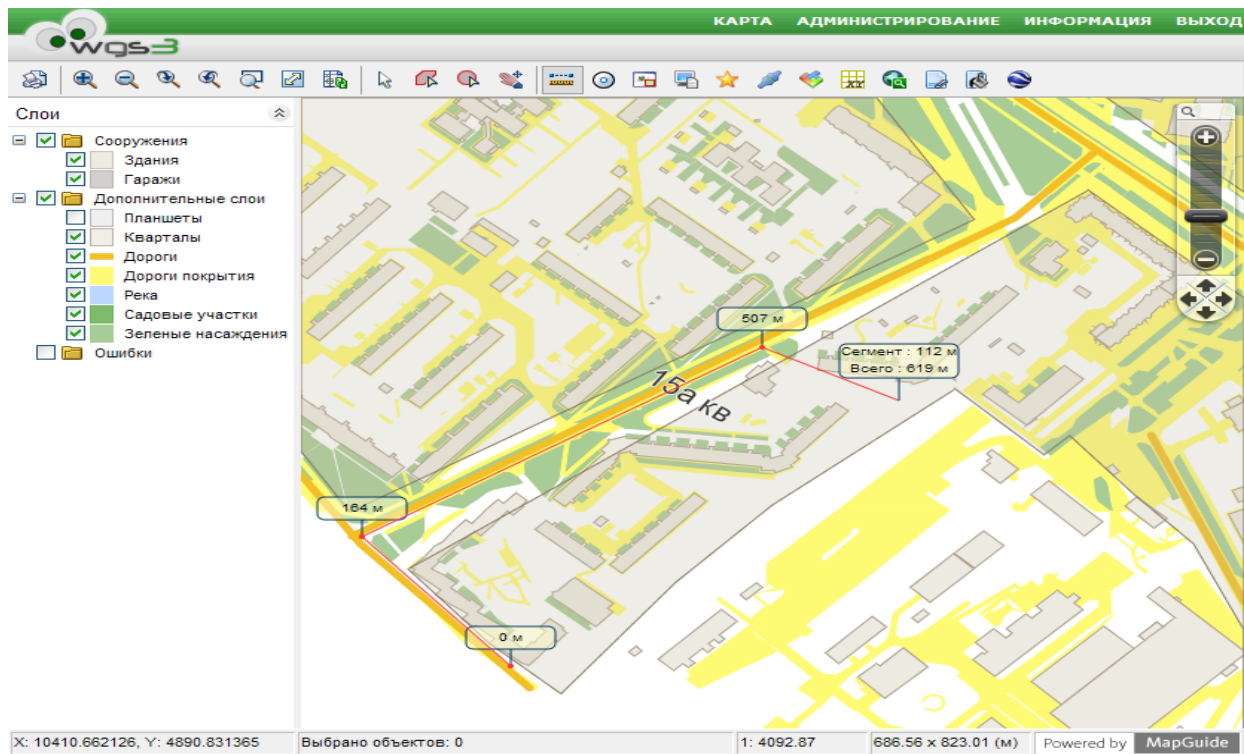


Рис. 5.22. Измерение расстояния



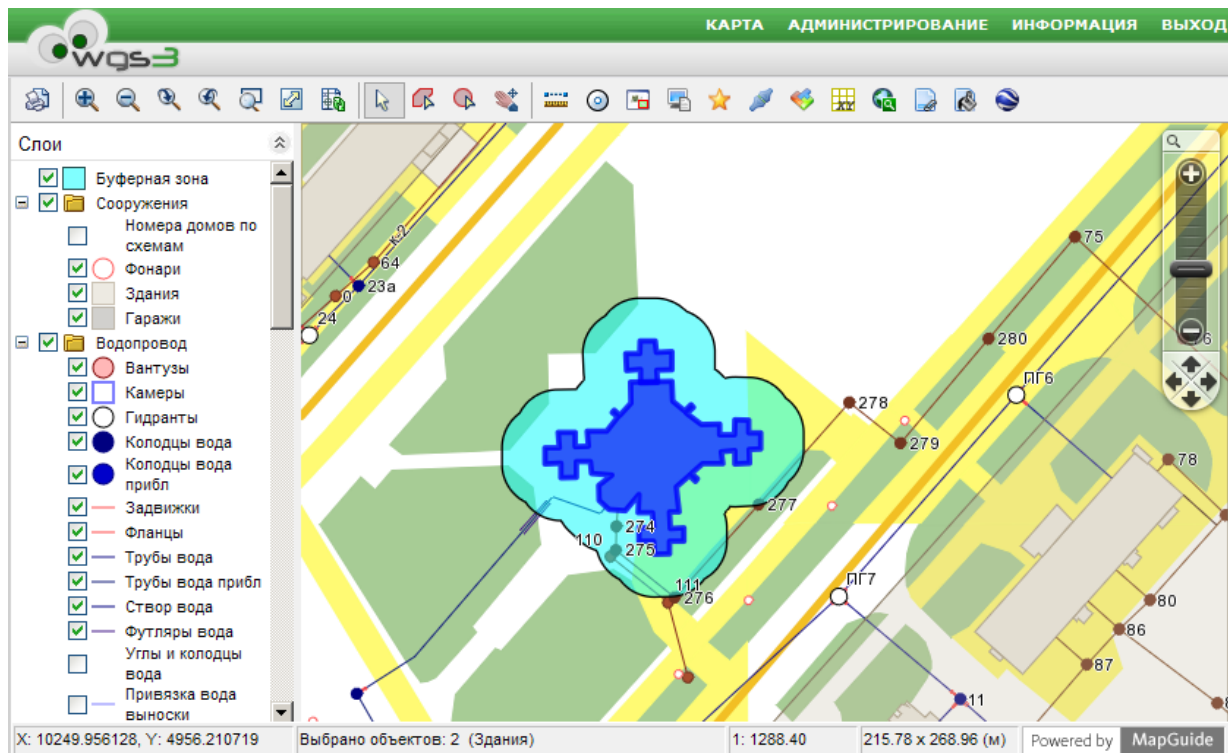


Рис. 5.23. Построение буферной зоны

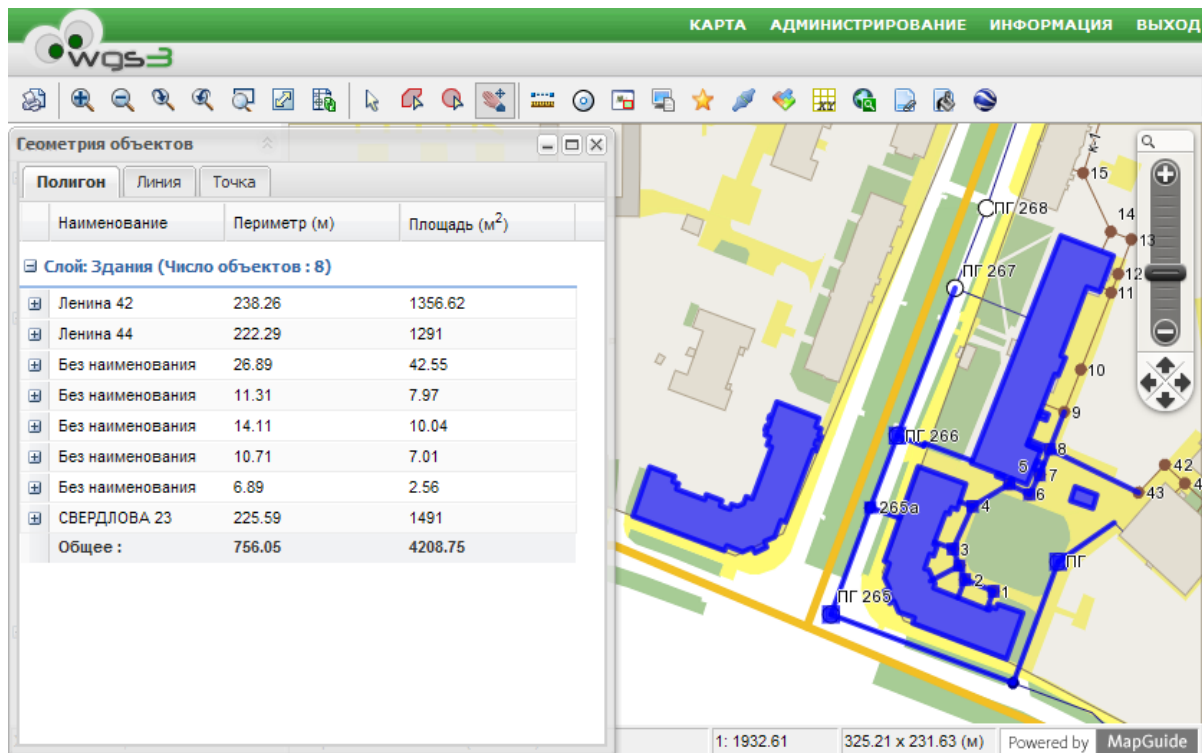


Рис. 5.24. Геометрия объектов

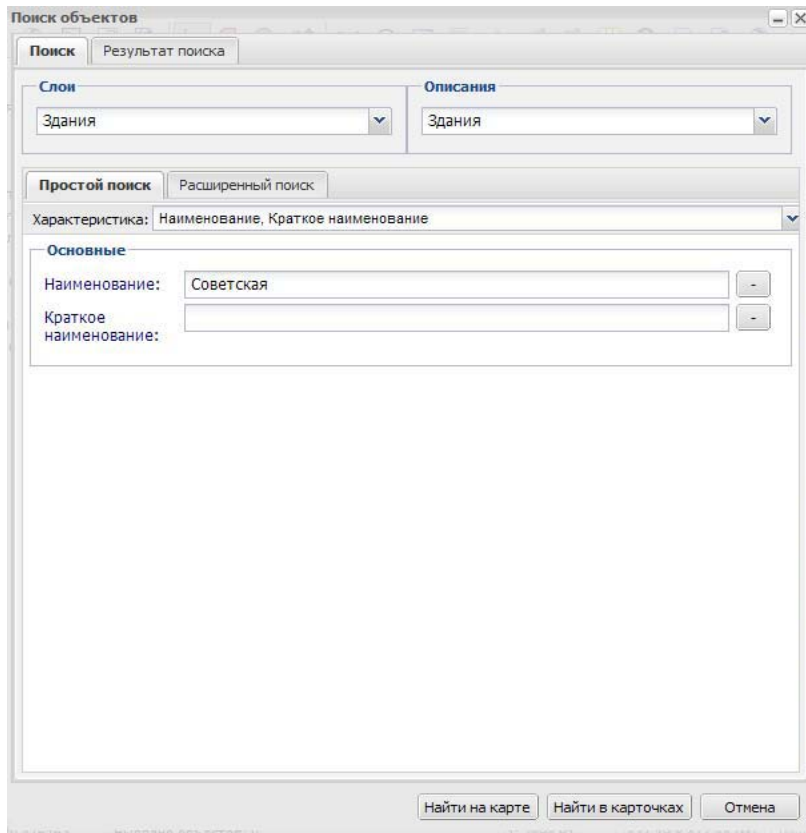


Рис. 5.25. «Простой» поиск объектов

Кнопка «Найти на карте» осуществляет поиск по основным и дополнительным карточкам (атрибутивным описаниям). Кнопка «Найти в карточках» осуществляет поиск по всем типам карточек.

**Расширенный поиск** (рис. 5.26) объектов предоставляет возможность создавать сложные запросы, позволяющие группировать условия, заключаемые в круглые скобки. Примером сложного запроса является, например, конструкция следующего типа: (характеристика1 = «значение1» ИЛИ характеристика2 содержит «значение2») И НЕ (характеристика3 < «значение3»).

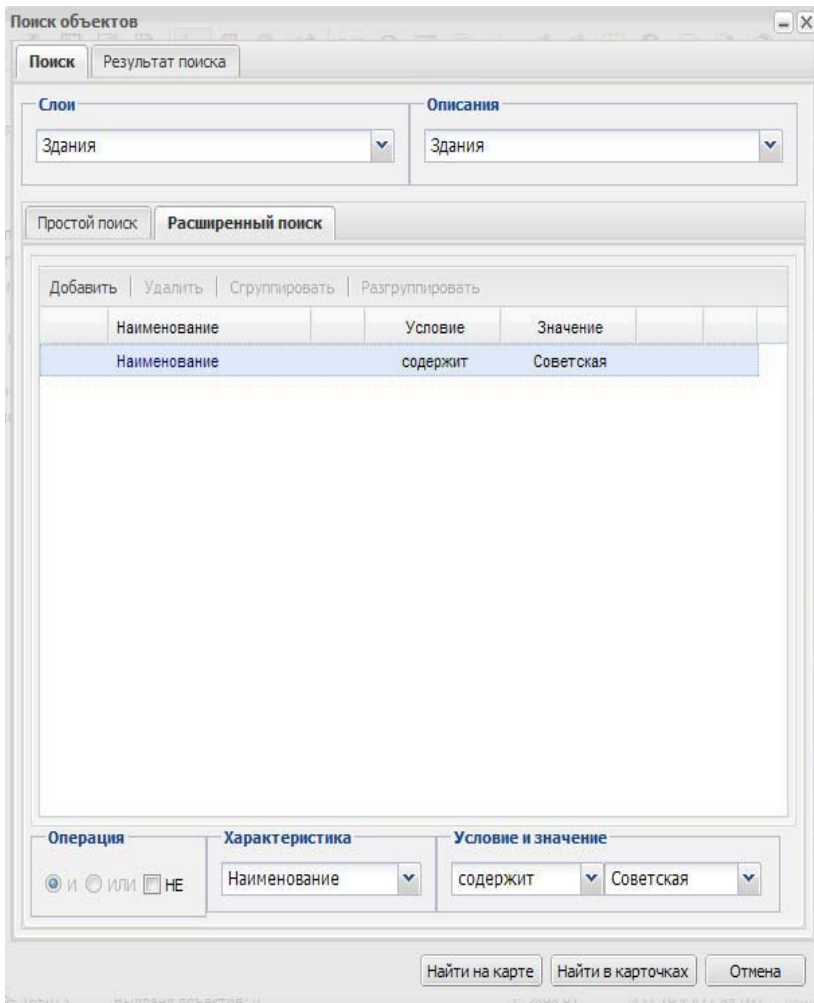


Рис. 5.26. Расширенный поиск объектов

Результаты поиска на карте изображены на рис. 5.27. Кнопка «Отобразить на карте» показывает объекты карты, соответствующие выбранным карточкам в таблице.

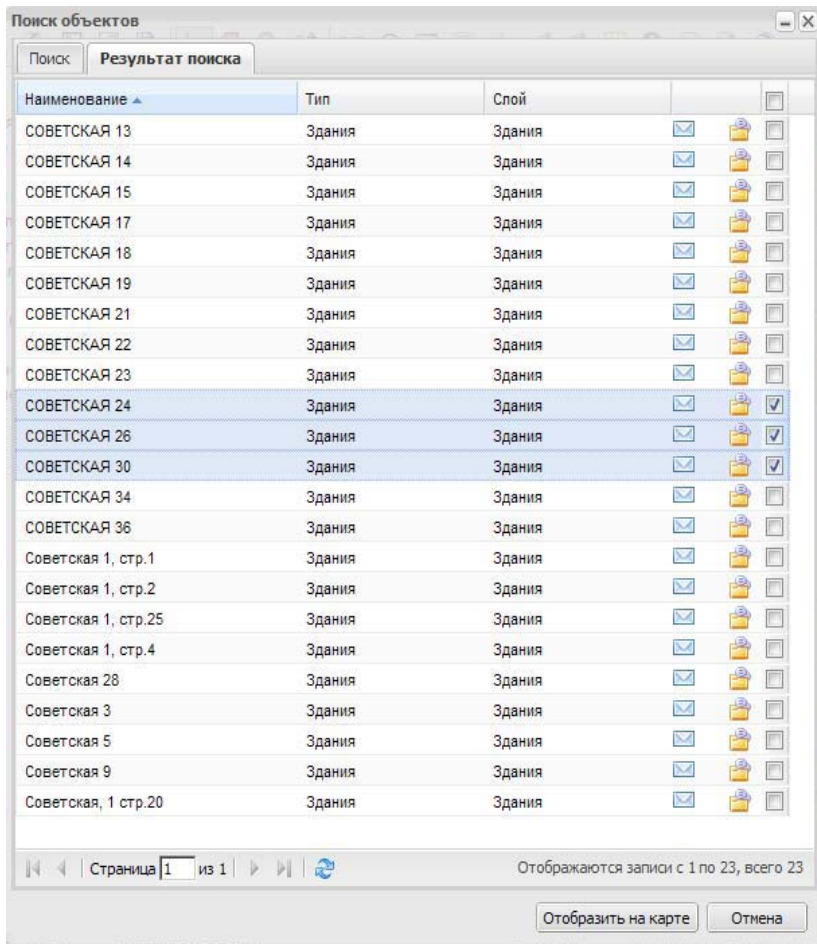


Рис. 5.27. Результаты поиска

**Карточка объекта** позволяет выводить информацию об объекте. После выбора одного объекта на карте появится форма, в которой отобразятся свойства объекта (рис. 5.28). При наличии прав на редактирование карточки возможно редактирование свойств.

The screenshot shows a window titled "Карточка" (Card) with a dropdown menu set to "Ленинградская 16 - Здания". The window is divided into several sections:

- Список описаний** (List of descriptions): A dropdown menu showing "Ленинградская 16 - Здания".
- Основные характеристики** (Main characteristics):
  - Наименование: Ленинградская 16
  - Краткое наименование: (empty)
  - Атрибутивное описание: Здания
  - Дата изменения: 2009-10-23 14:17:49
  - Автор: (empty)
  - Примечание: (empty text area)
- Группа характеристик: Общие** (Group of characteristics: General):
  - Назначение здания или сооружения: Магазин
  - Тип сооружения: 5юк
  - Дата постройки: 1976-09-10
  - Номер участка: 9
  - Строительный номер: 33
- Группа характеристик: Размеры и расстояния** (Group of characteristics: Dimensions and distances): (collapsed)

At the bottom of the window, there are four buttons: "Документы" (Documents), "Печать" (Print), "Применить" (Apply), and "Закрыть" (Close).

Рис. 5.28. Карточка объекта

При нажатии кнопки «Печать» формируется и открывается pdf-документ, который можно сохранить на диске или распечатать (рис. 5.29).

Печать

Save a copy (Shift+Ctrl+S)

Основные характеристики	
Наименование	Ленинградская 10
Краткое наименование	
Атрибутивное описание	Здания
Дата создания	2009-10-23 14:17:49
Примечание	

Группа характеристик : Общие	
Назначение здания или сооружения	Магазин
Тип сооружения	
Дата постройки	
Номер участка	
Строительный номер	

Группа характеристик : Размеры и расстояния	
Этажность над землей	0
Этажность под землей	
Состояние здания и сооружения	
Высота	
Объем отапливаемого помещения	
Площадь по внутреннему контуру	
Площадь застройки	
Периметр	

Группа характеристик : Адресные	
Улица	Ленинградская
Номер дома	16

Группа характеристик : Благоустройство	
Благоустройство вокруг здания	
Отмостка	
Польезные пути	

Группа характеристик : Технические	
Потребляемая мощность внутреннего оборудования	
Освещение (потребляемая мощность)	
Вентиляция (потребляемая мощность)	

Группа характеристик : Конструктивные	
Наличие мусоропроводов	
Тип фундамента	
Наличие электрификации	
Наличие канализации	

Рис. 5.29. Печать карточки объекта

Функция печати осуществляет **печать текущего фрагмента** карты. При выборе данной функции необходимо выбрать параметры печати: масштаб, формат бумаги, ориентацию листа и отображение легенды (рис. 5.30).

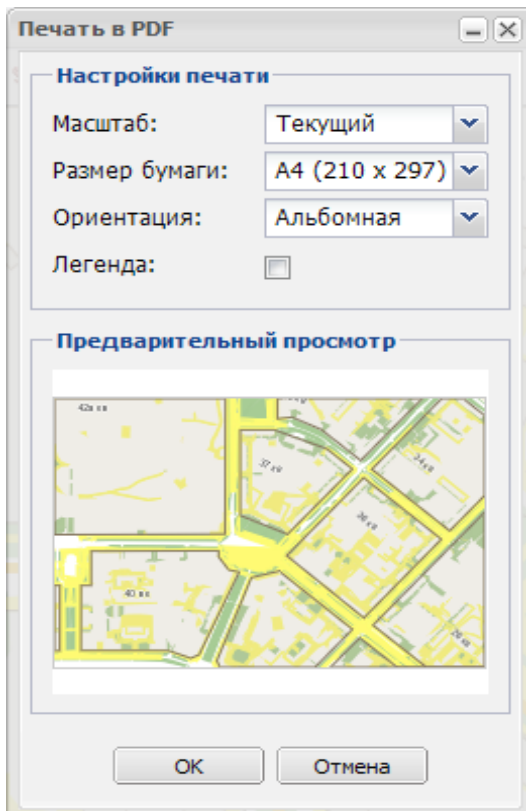


Рис. 5.30. Выбор параметров печати

При нажатии кнопки «ОК» (см. рис. 5.30) формируется файл с расширением pdf, который открывается с помощью плагина Acrobat Reader (рис. 5.31).



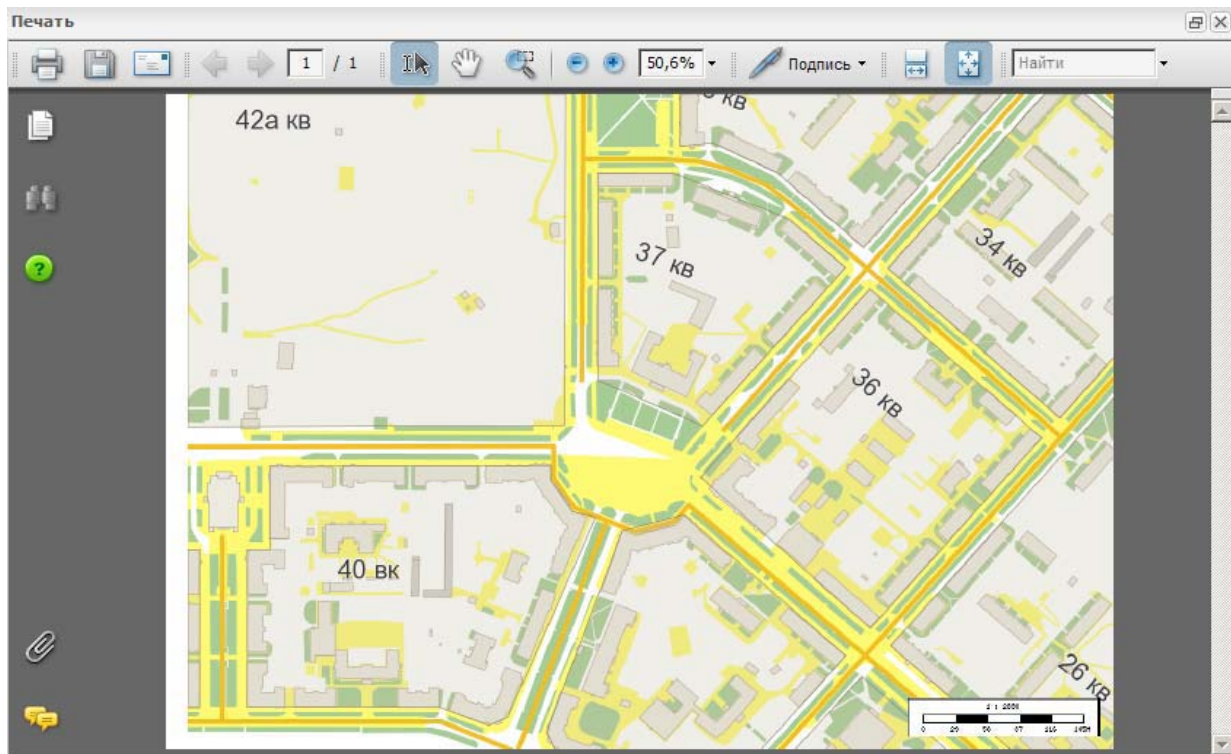


Рис. 5.31. Печать через плагин Acrobat Reader

1. Гриценко Ю.Б. Моделирование водопроводных сетей с использованием средств геоинформационных технологий: дис. ... канд. техн. наук. — Томск, 2000. — 141 с.
2. Евдокимов А.Г. Оперативное управление потокораспределением в инженерных сетях. — Харьков: Вища школа, 1980. — 144 с.
3. Мелентьев Л.А. Теплофикация. В 2 ч. — М.; Л.: АН СССР, 1944. — Ч. 1. — 248 с.; 1948. — Ч. 2. — 276 с.
4. Сарычев Д.С. Разработка информационно-расчетных комплексов для управления инженерными сетями и дорогами с использованием геоинформационных систем: дис. ... канд. техн. наук. — Томск, 2003. — 183 с.
5. Вайсфельд В.А., Ексаев А.Р. Принципиальные основы применения ГИС-технологий для городских инженерных коммуникаций // Инженерные коммуникации и геоинформационные системы: материалы первого учеб.-практ. семинара «ГИС-Ассоциация», 14–17 октября 1997 г. — М.: Изд-во «ГИС-Ассоциация», 1997. — С. 3–9.
6. Отчет по науч.-исслед. работе 1.2.01 «Исследование алгоритмов моделирования режимов функционирования территориально-распределенных инженерных сетей» / Ю.П. Ехлаков [и др.]. — М.: ВНИИЦ, 2005. — № ГР 02200505874.
7. Использование модели местности для анализа состояния окружающей среды / А.В. Скворцов [и др.] // Энергетика: экология, надежность, безопасность: материалы пятой Всерос. науч.-техн. конф. — Томск: Изд-во ТПУ, 1999. — С. 56
8. Кудинов А.В. Алгоритмическое и программное обеспечение геоинформационной системы для решения задач управления сетями инженерных коммуникаций: дис. ... канд. техн. наук. — Томск, 2002. — 176 с.
9. Болтянский В.Г., Ефремович В.А. Наглядная топология. — М.: Наука, 1983. — 160 с.
10. Фоменко А.Т. Наглядная геометрия и топология: Математические образы в реальном мире. — М.: Изд-во МГУ, 1988. — 416 с.
11. Королев Ю. К. Общая геоинформатика. Ч. I. Теоретическая геоинформатика. — М.: Изд-во СП ООО Дата+, 1998. — Вып. 1. — 118 с.
12. Цветков В.Я. Геоинформационные системы и технологии. — М.: Финансы и статистика, 1998. — 288 с.
13. Кошкарев А.В., Тикунов В.С. Геоинформатика / под ред. Д.В. Лищикова. — М.: Картгеоцентр; Геодезиздат, 1993. — 213 с.
14. Баранов Ю.Б., Бертлян А.М. Геоинформатика. Толковый словарь основных терминов. — М.: Изд-во ГИС-Ассоциации, 1999. — 204 с.

15. Капралов Е.Г., Коновалова Н.В. Введение в ГИС. — 2-е изд. — М.: Библион, 1997. — 160 с.

16. Развитие принципа масштабируемости приложений в среде геоинформационных систем / Ю.П. Ехлаков [и др.] // ГИС для оптимизации природопользования в целях устойчивого развития территорий: материалы Междунар. конф., 1–4 июля 1998 г. — Барнаул: Изд-во Алтайск. гос. ун-та, 1998. — С. 124–127.

17. Развитие средств математического моделирования инженерных сетей на базе геоинформационных технологий / Ю.П. Ехлаков [и др.] // ИНТЕРКАРТО–5. ГИС для устойчивого развития территорий: материалы Междунар. науч. конф. Ч. 2. Применение картографии и ГИС в науке и управлении систем. — Якутск: Изд-во Якутск. ун-та, 1999. — С. 88–96.

18. Гриценко Ю.Б., Жуковский О.И. Использование многоуровневой математической модели инженерной сети для геоинформационного мониторинга // Материалы Междунар. конф. SIBCONVERS'99, 18–20 мая 1999. Т. 2. — Томск: Томск. гос. ун-т систем управления и радиоэлектроники, 1999. — С. 328–330.

19. Гриценко Ю.Б., Панов А.С. Анализ представления структур данных для решения прикладных задач эксплуатации инженерных сетей // Доклады Томского государственного университета систем управления и радиоэлектроники. — 2004. — № 2 (10). — С. 26–30.

20. Принципы построения Web-ориентированной ГИС промышленного предприятия / Ю.П. Ехлаков [и др.] // Известия Томского политехнического университета. — 2006. — Т. 309, № 7. — С. 146–152.

21. Веб-ГИС // Компьютерраonline: электронный журнал. — 2008. — № 33. — URL: <http://offline.computerra.ru/2008/749/371966/> (дата обращения: 24.11.2010).

22. Прикладная геоинформатика / А.Д. Иванников [и др.]. — М.: МАКС Пресс, 2005. — 360 с.

23. Применение технологии шаблонов при создании Web-приложений / О.И. Жуковский [и др.] // Электронные средства и системы управления: тез. докл. Всерос. науч.-практ. конф., 21–23 октября 2003 г. — Томск: ГНУ «НИИ АЭМ при ТУСУР», 2003. — С. 147–152.

24. Проблема идентификации пространственных объектов / В.Ю. Вишняков [и др.] // Научная сессия ТУСУР-2007: материалы докл. Всерос. науч.-техн. конф. студентов, аспирантов и молодых ученых. Тематический выпуск «Системная интеграция и безопасность», Томск, 3–7 мая 2007 г. В 2 ч. Ч. 1. — Томск: В-Спектр, 2007. — С. 347–350.

25. Технология хеширования пространственных данных / В.Ю. Вишняков [и др.] // Современная техника и технологии: тр. XIII Междунар.

науч.-практ. конф. студентов, аспирантов и молодых ученых, 26–30 марта 2007 г. В 3 т. Т. 2. — Томск: Изд-во ТПУ, 2007. — С. 303–305.

26. Решение проблемы обновления пространственных данных в среде Oracle Spatial / В.Ю. Вишняков [и др.] // Доклады Томского государственного университета систем управления и радиоэлектронники. — 2008. — № 2 (18). — С. 76–79.

27. Вишняков В.Ю., Гриценко Ю.Б. Модель данных для представления топологических связей в инженерных сетях // SIBCON-2007: материалы Междунар. конф. — Томск: Томск. гос. ун-т систем управления и радиоэлектроники, 2007. — С. 89–93.

28. Репозиторий RFC на русском языке. — URL: <http://rfc.com.ru/rfc1321.htm> (дата обращения: 23.03.2010).

29. Проблемы представления пространственной информации в web-ориентированных геоинформационных системах / Ю.Б. Гриценко [и др.] // Инновационные недра Кузбасса. IT-технологии: сб. науч. тр. — Кемерово: ИНТ, 2007. — С. 215–216.

30. Фаулер М. Архитектура корпоративных программных приложений / пер. с англ. — К.; М.; СПб: Вильямс, 2004. — 544 с.

31. Жуковский О.И., Рыбалов Н.Б. Архитектура корпоративной WEB-ориентированной ГИС // Доклады Томского государственного университета систем управления и радиоэлектроники. — 2008. — № 2 (18). Ч. 2. — С. 46–57.

32. Проблемы организации клиентской части в web-ориентированных геоинформационных системах / Ю.Б. Гриценко [и др.] // Электронные средства и системы управления. Опыт инновационного развития: докл. Междунар. науч.-практ. конф., 31 октября – 3 ноября 2007 г. В 2 ч. Ч. 2. — Томск: В-Спектр, 2007. — С. 251–253.

# Оглавление

Введение .....	3
<b>Глава 1. Представление инженерных сетей</b>	
1.1. Типология инженерных сетей и классификация задач эксплуатации .....	5
1.2. Моделирование топологических отношений в инженерных сетях .....	19
<b>Глава 2. Моделирование инженерных сетей</b>	
2.1. Алгебраическая модель инженерной сети .....	28
2.2. Унификация параметров и функций алгебраической модели инженерной сети .....	40
2.3. Представления структур данных для решения задач моделирования инженерных сетей .....	48
<b>Глава 3. Технологии функционирования инженерных ГИС</b>	
3.1. История использования ГИС .....	57
3.2. Информационные технологии построения веб-ориентированной ГИС .....	61
3.3. Технология хеширования данных .....	71
3.4. Технология разграничения прав доступа к векторным объектам .....	76
<b>Глава 4. Архитектуры веб-ориентированной ГИС</b>	
4.1. Многоуровневая архитектура веб-ориентированных ГИС	81
4.2. Создание эффективного пользовательского интерфейса	83
4.3. Выбор архитектуры приложения на стороне сервера .....	86
4.4. Выбор архитектуры приложения на стороне клиента .....	92
4.5. Сервис-ориентированный подход при построении геоинформационной системы .....	94
4.6. Построение веб-ориентированной ГИС как Mashup-приложения .....	99
4.7. Сервис-ориентированная архитектура блока анализа веб-ориентированной ГИС .....	103
<b>Глава 5. Веб-ориентированная геоинформационная         система WGS3</b>	
5.1. Администрирование. Управление проектом .....	109
5.2. Администрирование. Управление данными .....	114
5.3. Пользовательские функции системы .....	128
Литература .....	145