

ОБЩАЯ ХАРАКТЕРИСТИКА РАБОТЫ	4
1. АНАЛИЗ СУЩЕСТВУЮЩИХ ИНФОРМАЦИОННЫХ СИСТЕМ МОНИТОРИНГА И УПРАВЛЕНИЯ ПРОЕКТАМИ	10
1.1. Microsoft Project	10
1.2. ProjectMate	12
1.3. OpenProj	13
1.4. Автоматизированная система принятия решения и объектного мониторинга мероприятий, проводимых на территории региона	14
1.5. Результаты анализа существующих аналогов . . .	15
2. МОДЕЛЬ СИСТЕМЫ ПОДДЕРЖКИ ПРИНЯТИЯ РЕШЕНИЙ И МОНИТОРИНГА ПРОЕКТОВ . . .	17
2.1. Классификация систем поддержки принятия решений	17
2.2. Архитектура систем поддержки принятия решений	21
2.2.1. Функциональная СППР	24
2.2.2. СППР с использованием независимых витрин данных	25
2.2.3. СППР на основе двухуровневого хранилища данных	27
2.2.4. СППР на основе трёхуровневого хранилища данных	28
2.3. Методы интеллектуального анализа данных (Data Mining)	31

2.4.	Архитектура кроссплатформенной системы под- держки принятия решений	33
2.5.	Описание алгоритма работы менеджера загрузки	37
3.	ОПИСАНИЕ КРОССПЛАТФОРМЕННОЙ СИСТЕ- МЫ ПОДДЕРЖКИ ПРИНЯТИЯ РЕШЕНИЙ И МОНИТОРИНГА ПРОЕКТОВ	40
3.1.	Диаграмма классов предметной области	40
3.2.	Архитектура приложения и инструменты разра- ботки	43
3.3.	Экспериментальная проверка алгоритмов	47
3.4.	Функции системы	49
3.5.	Перспективы развития	53
	ЗАКЛЮЧЕНИЕ	55
	БИБЛИОГРАФИЧЕСКИЙ СПИСОК	56
	ПРИЛОЖЕНИЕ 1. Материалы на электронном носителе	64

ОБЩАЯ ХАРАКТЕРИСТИКА РАБОТА

Актуальность темы. В настоящее время наблюдаются тенденции возрастания сложности проектов, и, как следствие, повышения трудоемкости учета их реализации на всех этапах жизненного цикла.

Достоверность, не противоречивость и полнота проектной информации напрямую влияет на качество управленческих решений и своевременность их принятия. Этот аспект обуславливает актуальность построения системы управления на базе современных технологий автоматизации процессов информационной поддержки принятия решений [1]. Данное обстоятельство приводит к необходимости ведения учета реализации проектов с использованием информационных технологий ведомствами, относящимися к различным сферам деятельности. В данном случае, объектом применения информационных технологий являются процессы сбора, обработки, хранения, анализа, представления и визуализации информации.

Обобщенный механизм мониторинга проектной деятельности показан на рисунке 1. Каждое из ведомств ведет учет своих собственных проектов. Управляющее ведомство отвечает за координацию работ по всем проектам всех подчиненных ведомств. Своевременное получение актуальной и достоверной информации о состоянии дел в подконтрольных ведомствах позволяет повысить качество управленческих решений для управляющего ведомства. Очевидно, что

информация о ведении проектов различными ведомствами имеет общую специфику и можно выделить некоторые основные критерии отбора информации из различных подсистем для анализа и последующего представления управляющему ведомству.

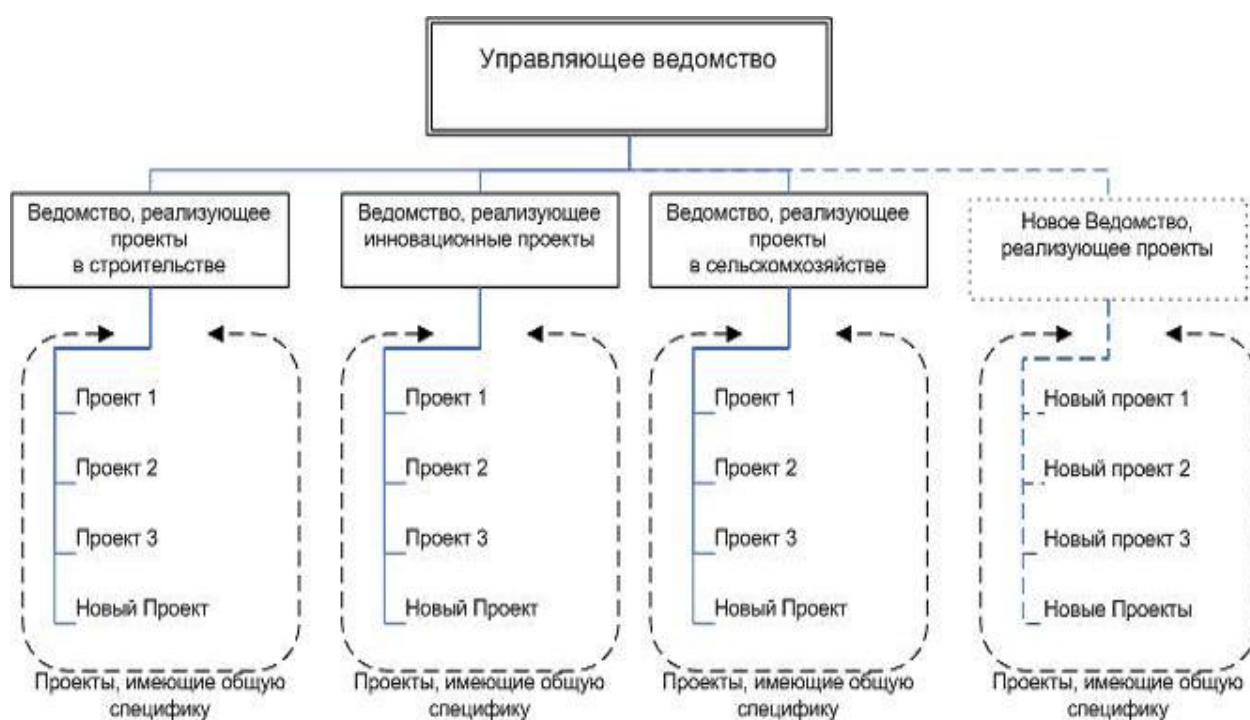


Рисунок 1. Обобщенный механизм мониторинга проектной деятельности

В действительности управлением и мониторингом проектной деятельности могут заниматься различные подразделения с неравномерным распределением ответственности между ними (рисунок. 2). Например, за различные этапы проекта ответственными за реализацию могут быть назначены представители разных ведомств. Также проекты могут иметь обширное пространственное представление и определенные объекты проектов могут относиться к различным географическим объектам, что и показано на диаграмме на примере муниципальных

образований. Реализация некоторых проектов может охватывать достаточно обширные территории: проект может включать в себя несколько пространственно разрозненных объектов, на которых ведутся работы, или сами объекты могут иметь не тривиальную форму представления на географической карте (например, быть полигоном или иметь протяженную форму). Существующие реалии нужно учитывать при разработке информационных средств мониторинга проектной деятельности.

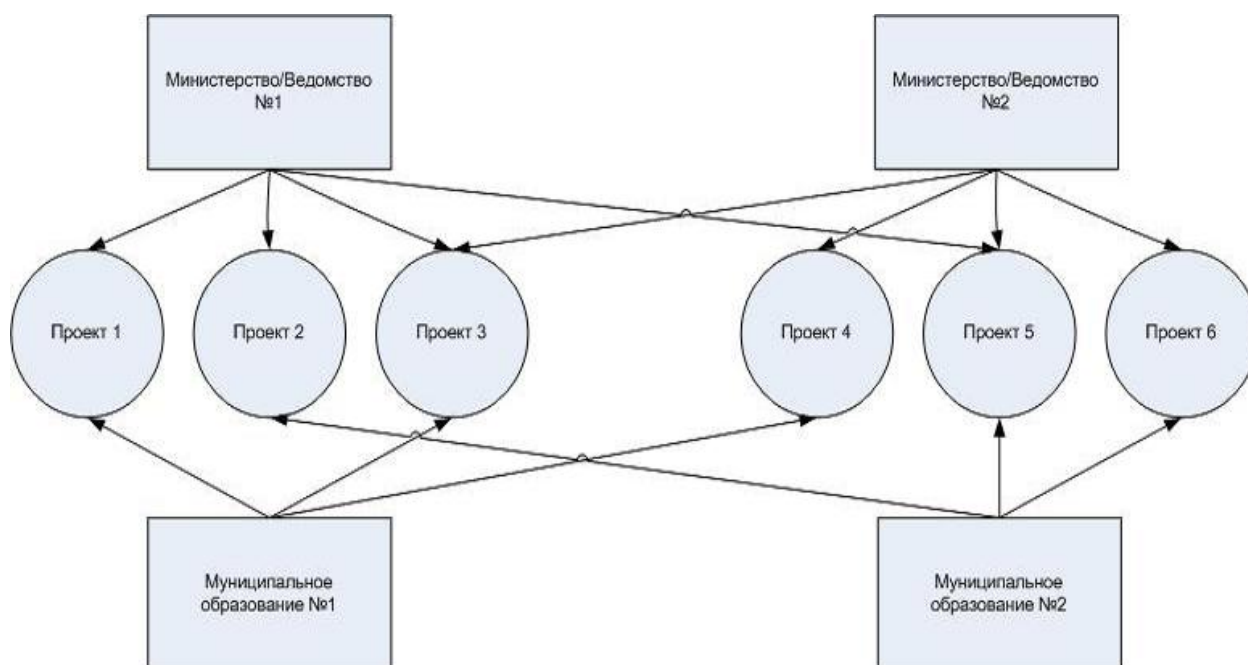


Рисунок 2. Детализированный механизм мониторинга проектной деятельности

Современный рынок информационных технологий предлагает набор информационных систем для решения подобных задач. Но большая часть данных систем разрабатывалась для ведения учета конкретных категорий проектов и с учетом требований конкретных ведомств.

С ростом числа ведомств, использующих системы управления проектами, для учета и отслеживания процесса

реализации выполняемых ими проектов появилась необходимость более глобального учета деятельности всех ведомств, что привело к необходимости централизации управления ведомственными и межведомственными проектами. Таким образом, становится актуальной разработка автоматизированной системы мониторинга выполнения проектов, позволяющей вести полноценный контроль выполнения работ. Для получения актуальной и своевременной информации по реализации проектов ведомствами необходимо консолидировать информацию из ведомственных подсистем с учетом их отраслевых специфик и особенностей.

Кроме того, для визуализации статистической информации, связанной с ведением проектов, целесообразно использование картографических данных и, как следствие, взаимодействовать с геоинформационной системой.

В настоящее время наиболее оптимальным решением задач по учету и мониторингу ведения проектов различными структурами является привлечение услуг специализированных предприятий или аутсорсинг. На данный момент достаточно большое число компаний предлагают подобные услуги. Данные решения имеют ряд недостатков, таких как необходимость заключения долгосрочного контракта с аутсорсинговой компанией, что неизбежно влечет дополнительные финансовые затраты, которые могут стать необоснованными в случае расторжения контракта, а также жесткую привязку к сложившемуся формату отношений.

В такой ситуации может стать актуальным использование

информационных технологий для реализации функционала аутсорсинговой компании, что может стать более универсальным решением задачи мониторинга ведения проектов различными структурами.

Цель диссертации. Повышение эффективности принятия управленческих решений за счет получения актуальной, достоверной аналитической и статистической информации по учету ведения проектов.

Основные задачи, которые потребовалось решить для достижения поставленной цели:

- проанализировать существующие автоматизированные системы поддержки принятия решений и мониторинга проектов и сформулировать требования, которым должна удовлетворять система;
- спроектировать архитектуру системы;
- разработать механизмы взаимодействия с внешними разнородными информационными системами;
- провести исследование существующих математических методов анализа фактографической информации;
- выбрать необходимые средства разработки;
- создать кроссплатформенную информационно-аналитическую систему управления проектами.

Научная новизна результатов, полученных в диссертации, заключается в следующем:

- создана модель кроссплатформенной системы, отличающейся возможностью консолидации

информации из разнородных источников данных о проектной деятельности;

- разработан оригинальный алгоритм интеграции информации из разнородных баз данных.

Методы исследований. Представленные в диссертационной работе научные положения обоснованы с применением методологии структурного и системного анализа информационных систем, теории реляционных баз данных, технологии построения систем баз данных, технологии программирования, а также математических методов анализа информации.

ГЛАВА 1. АНАЛИЗ СУЩЕСТВУЮЩИХ ИНФОРМАЦИОННЫХ СИСТЕМ МОНИТОРИНГА И УПРАВЛЕНИЯ ПРОЕКТАМИ

Анализ существующих решений на рынке производителей программного обеспечения выявил ряд информационных систем, предназначенных для автоматизации процессов мониторинга проектной деятельности и поддержки принятия управленческих решений. Среди них присутствуют как зарубежные, так и отечественные разработки.

Существующие программные продукты, включающие в себя приложения для планирования задач, составления расписания, контроля цены и управления бюджетом, распределения ресурсов, совместной работы, общения, быстрого управления, документирования и администрирования системы, используются совместно для управления крупными проектами. К этим системам относятся:

- Microsoft Project;
- ProjectMate;
- OpenProj;
- Автоматизированная система принятия решения и объектного мониторинга мероприятий, проводимых на территории региона;

1.1. Microsoft Project

Программный комплекс по управлению проектной деятельностью, разработанный корпорацией Microsoft. Под маркой Microsoft Project доступны сразу несколько продуктов

и решений [3]:

- Microsoft Project Standard — однопользовательская версия для небольших проектов;
- Microsoft Project Professional — корпоративная версия продукта поддерживающая совместное управление проектами и ресурсами, а также управление портфелями проектов с помощью Microsoft Project Server;
- Microsoft Project Web Access — Web-интерфейс для отчетности о выполнении задач, а также просмотра портфелей проектов;
- Microsoft Project Portfolio Server — продукт для отбора проектов для запуска на основе сбалансированных показателей.

Продукты различаются между собой по своему функционалу, возможности сетевой работы с проектами, интеграции с Microsoft SQL Server и Microsoft SharePoint. Стоимость одной копии с возможностью совместной работы над проектам составляет порядка 30000 руб.

Корпоративная версия Microsoft Project требует квалифицированного персонала для внедрения, сопровождения и работы. Microsoft ведет ряд сертификационных программ по MS Project для проверки квалификации аналитиков. Стоимость подготовки менеджера корпоративных проектов в Microsoft Project Professional 2010/2007 составляет порядка 20000–25000 руб.

Решение на базе Microsoft Project Professional обладает функциональностью для организации коллективного ведения и планирования проектной деятельности, построения различного вида диаграмм и графиков, учета и прогнозирования ресурсов и многого другого - в целом данный продукт обладает широкими возможностями. Но, учитывая специфику поставленной задачи, большая часть представленного функционала данного ПО оказывается невостребованной.

С другой стороны MS Project не предоставляет возможности интеграции агрегированной информации о проектной активности из разнородных источников, использующих различные системы управления базами данных и имеющих разную реляционную структуру организации хранения данных, а также в нем отсутствует геоинформационная составляющая.

1.2. ProjectMate

Российская система автоматизации профессиональной деятельности, относящаяся к классу PSA-решений (Professional Services Automation), разрабатываемая компанией «Авиком Бизнес Технологии» [4].

ProjectMate включает в себя следующие модули [5]:

- управление временем и затратами;
- управление проектами и ресурсами;
- управление финансами;
- управление запросами;
- управление контактами;
- документооборот.

ProjectMate позволяет вести детальный учет времени по всем сотрудникам и проектам, планировать и контролировать исполнение бюджетов проектов, осуществлять сбор информации о всех выполняемых проектах и задействованных в них ресурсах, получать отчеты о коммерческой деятельности и др.

Стоимость лицензии на сервер Avicom ProjectMate с правом использования в течение одного года составляет порядка 16000 руб. Одна лицензия на индивидуального пользователя на тот же срок редакции Professional — 9950 руб.

Очевидным недостатком данной системы является привязанность к программной платформе на базе операционной системы Microsoft Windows.

1.3. OpenProj

Кроссплатформенное программное обеспечение для управления проектами, разрабатываемое компанией Serena Software. OpenProj позволяет пользователям:

- строить диаграммы Ганта;
- создавать сетевые графики;
- вести управление проектными ресурсами;
- генерировать различные виды отчетов;
- фильтровать, сортировать и группировать задачи по заданным критериям;
- конструировать иерархические структуры связанных работ.

Следует отметить, что OpenProj распространяется под свободной лицензией Common Public Attribution License

Version 1.0 (CPAL) и поддерживает операционные системы Linux, Unix, Mac OS X и Microsoft Windows.

Существенным недостатком является отсутствие возможности сетевой работы пользователей над проектами.

1.4. Автоматизированная система принятия решения и объектного мониторинга мероприятий, проводимых на территории региона

Данная информационная система состоит из двух компонент: подсистемы интерактивной обработки информации по учету и контролю проектной деятельности и web-ориентированного модуля, предназначенного для осуществления мониторинга временных и финансовых ресурсов на текущем этапе проведения проектных работ [2].

Отличительными функциональными возможностями системы являются:

- возможность ведения проектов межведомственного характера;
- фильтрация проектов по критериям: мероприятиям, объектам, этапам и др.
- построение графиков и диаграмм с учетом фильтрации по типам проектов;
- предоставление аналитической информации в виде отчетов;
- интеграция с географической электронной картой г. Астрахань с возможностью навигации и получения информации по объектам проектной деятельности.

Рассматриваемая система имеет широкие возможности по

организации коллективной работы над проектами, а также поддерживает разграничение прав доступа к информации для различных типов пользователей.

С другой стороны, отсутствие возможности интеграции информации из различных источников данных и ориентированность подсистемы обработки информации на Windows платформу являются значительными недостатками.

1.5. Результаты анализа существующих аналогов

Сравнительная таблица функциональных возможностей рассмотренных аналогов разрабатываемой информационной системе приведена в таблице 1.1.

Таблица 1.1

Сравнительная таблица аналогов

Название	1	2	3	4	5
Microsoft Project	+	—	+	—	—
ProjectMate	+	—	+	—	—
OpenProj	+	+	—	—	—
Автоматизированная система принятия решения и объектного мониторинга мероприятий, проводимых на территории региона	+	—	+	—	+

Перечень критериев, используемых в таблице 1.1.

- 1 — русификация;
- 2 — кроссплатформенность;
- 3 — сетевая работа;

- 4 — работа с различными источниками данных;
- 5 — интеграция с картографическим сервером.

Проведенный анализ показал, что программное обеспечение сторонних разработчиков удовлетворяет трем параметрам из пяти необходимых. К тому же ни одна из проанализированных информационных систем не имеет функциональной возможности работы с несколькими независимыми источниками данных, которые могут находиться под управлением различных СУБД и иметь различную структуру организации данных. Поддержку геоинформационной составляющей из всех рассмотренных имеет только автоматизированная система принятия решения и объектного мониторинга мероприятий, проводимых на территории региона.

На основе полученных данных можно сделать вывод, что что данные системы предназначены в первую очередь для непосредственного управления и мониторинга конкретного проекта, но не позволяют организовать централизованную консолидацию данных о процессе реализации разнотипных проектов, а также организовать интеграцию с системами управления конкретными проектами, реализованных на различных программных платформах.

Учитывая результаты проведенного исследования, приходим к выводу, что разработка кроссплатформенной системы поддержки принятия решений и мониторинга проектов, в которой будет реализован заявленный выше функционал, актуальна.

ГЛАВА 2. МОДЕЛЬ СИСТЕМЫ ПОДДЕРЖКИ ПРИНЯТИЯ РЕШЕНИЙ И МОНИТОРИНГА ПРОЕКТОВ

2.1. Классификация систем поддержки принятия решений

Современные системы поддержки принятия решения (СППР) представляют собой системы, максимально приспособленные к решению задач повседневной управленческой деятельности, являются инструментом, призванным оказать помощь лицам, принимающим решения (ЛПР). С помощью СППР может производиться выбор решений некоторых неструктурированных и слабоструктурированных задач, в том числе и многокритериальных [9].

СППР, как правило, являются результатом мультидисциплинарного исследования, включающего теории баз данных, искусственного интеллекта, интерактивных компьютерных систем, методов имитационного моделирования.

СППР — в большинстве случаев — это интерактивная автоматизированная система, которая помогает пользователю использовать данные и модели для идентификации и решения задач и принятия решений. Система должна обладать возможностью работать с интерактивными запросами с достаточно простым для изучения языком запросов [31].

Для СППР отсутствует не только единое общепринятое

определение, но и исчерпывающая классификация. Разные авторы предлагают разные классификации.

На уровне пользователя СППР разделяют на пассивные, активные и кооперативные. Пассивной СППР называется система, которая помогает процессу принятия решения, но не может вынести предложение, какое решение принять. Активная СППР может сделать предложение, какое решение следует выбрать. Кооперативная позволяет ЛПР изменять, пополнять или улучшать решения, предлагаемые системой, посылая затем эти изменения в систему для проверки. Система изменяет, пополняет или улучшает эти решения и посылает их опять пользователю. Процесс продолжается до получения согласованного решения.

На концептуальном уровне выделяют СППР [10]:

- управляемые сообщениями (Communication-Driven DSS);
- управляемые данными (Data-Driven DSS);
- управляемые документами (Document-Driven DSS);
- управляемые знаниями (Knowledge-Driven DSS);
- управляемые моделями (Model-Driven DSS).

СППР, управляемые моделями, характеризуют в основном доступ и манипуляции с математическими моделями (статистическими, финансовыми, оптимизационными, имитационными). Некоторые OLAP-системы, позволяющие осуществлять сложный анализ данных, могут быть отнесены к гибридным СППР, которые обеспечивают моделирование, поиск и обработку данных.

Управляемая сообщениями (Communication-Driven DSS) СППР поддерживает группу пользователей, работающих над выполнением общей задачи.

СППР, управляемые данными (Data-Driven DSS) или СППР, ориентированные на работу с данными (Data-oriented DSS или Business Intelligence) в основном ориентируются на доступ и манипуляции с данными.

СППР, управляемые документами (Document-Driven DSS), управляют, осуществляют поиск и манипулируют неструктурированной информацией, заданной в различных форматах.

СППР, управляемые знаниями (Knowledge-Driven DSS) обеспечивают решение задач в виде фактов, правил, процедур.

На техническом уровне различают СППР всего предприятия и настольную СППР [11]. СППР всего предприятия подключена к большим хранилищам информации и обслуживает многих менеджеров предприятия. Настольная СППР — это малая система, обслуживающая лишь один компьютер пользователя.

В зависимости от данных, с которыми эти системы работают, СППР условно можно разделить на оперативные и стратегические. Оперативные СППР предназначены для немедленного реагирования на изменения текущей ситуации в управлении финансово-хозяйственными процессами компании. Стратегические СППР ориентированы на анализ значительных объемов разнородной информации, собираемых из различных источников. Важнейшей целью этих СППР является поиск

наиболее рациональных вариантов развития бизнеса компании с учетом влияния различных факторов, таких как конъюнктура целевых для компании рынков, изменения финансовых рынков и рынков капиталов, изменения в законодательстве и др. СППР первого типа получили название Информационных Систем Руководства (Executive Information Systems, ИСР). По сути, они представляют собой конечные наборы отчетов, построенные на основании данных из транзакционной информационной системы предприятия, в идеале адекватно отражающей в режиме реального времени основные аспекты производственной и финансовой деятельности. Для ИСР характерны следующие основные черты:

- отчеты, как правило, базируются на стандартных для организации запросах, число последних относительно невелико;
- ИСР представляет отчеты в максимально удобном виде, включающем, наряду с таблицами, деловую графику, мультимедийные возможности и т. п.;
- как правило, ИСР ориентированы на конкретный вертикальный рынок, например финансы, маркетинг, управление ресурсами.

СППР второго типа предполагают достаточно глубокую проработку данных, специально преобразованных так, чтобы их было удобно использовать в ходе процесса принятия решений. Неотъемлемым компонентом СППР этого уровня являются правила принятия решений, которые на основе агрегированных данных дают возможность менеджерам

компании обосновывать свои решения, использовать факторы устойчивого роста бизнеса компании и снижать риски. СППР второго типа в последнее время активно развиваются. Технологии этого типа строятся на принципах многомерного представления и анализа данных (OLAP).

Проектируемая СППР в соответствии с предъявленными требованиями относится к типу пассивных и стратегических систем, так как ориентирована на анализ значительного количества информации, содержащейся в разнородных источниках данных.

2.2. Архитектура систем поддержки принятия решений

Архитектура перспективных систем поддержки принятия решений уровня всего предприятия представлена на рисунке 2.1 [12].

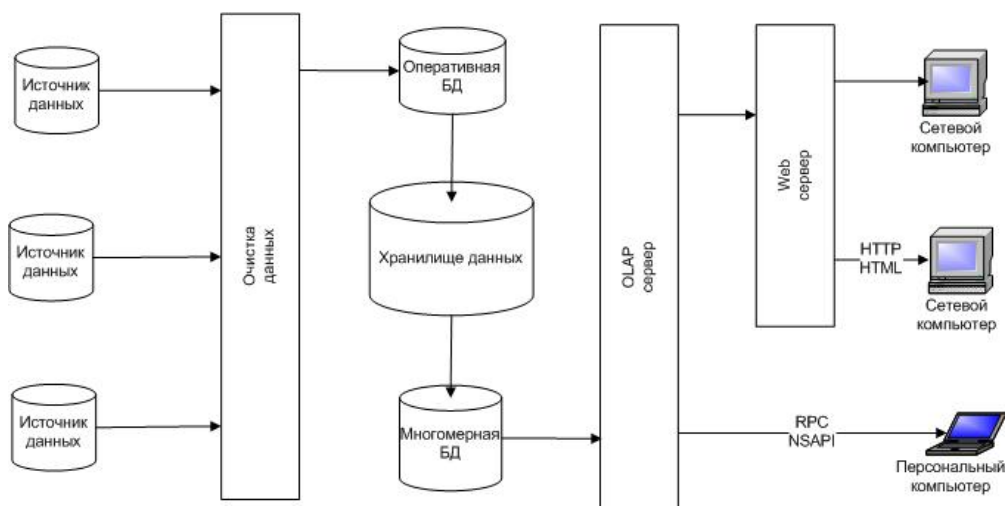


Рисунок 2.1. Архитектура перспективных систем поддержки принятия решений

В архитектуре перспективных СППР можно выделить 3 базовых подсистемы:

- ввода, очистки и преобразования данных;
- хранения данных;

- анализа данных.

Подсистема ввода данных занимается загрузкой данных из различных источников данных, которыми обычно являются OLTP (Online transaction processing) системы, занимающиеся операционной (транзакционной) обработкой данных. Известно, что оперативные системы обработки данных характеризуются очень низким уровнем качества данных с точки зрения их применения для поддержки принятия решений, а также высокой вероятностью появления ошибочных данных. При выгрузке данных из нескольких не взаимосвязанных между собой источников информации возникают проблемы синхронизации и логического сопоставления данных (например, Иванов И.И. в базе данных экономического отдела и Иванов И.И. в базе отдела кадров — это один и тот же человек). В этой связи подсистема ввода данных является важнейшим ключевым звеном в архитектуре СППР, от работы которого зависит корректность загружаемых данных и, как следствие, качество принимаемых стратегических решений.

Подсистема организации хранения данных занимается вопросами организации обработанных данных для дальнейшего упрощения работы с ними. Для реализации данной подсистемы используют современные системы управления базами данных и концепцию хранилищ данных. Также оправданным является использование многомерных баз данных. В системах анализа данных обычно требуется выполнение операций сразу над большими объемами данных с широким применением операций группировки и обобщений.

Поэтому возможны обобщение и агрегация данных, а также предрасчет некоторых показателей и помещение их в хранилище данных.

Подсистема анализа данных выполняет функции непосредственного анализа и обработки информации. Данная подсистема может включать в себя подсистемы:

- информационного-поискового анализа на базе РСУБД и статических запросов;
- оперативного анализа с применением технологии OLAP на базе концепции многомерного представления данных;
- интеллектуального анализа, реализующего методы и алгоритмы Data Mining.

Подсистема анализа данных может включать в себя широкое множество математических методов и алгоритмов обработки и анализа данных, а также алгоритмы прогнозирования развития ситуаций и организации активных СППР [?].

На сегодняшний день можно выделить четыре наиболее популярных типа архитектур систем поддержки принятия решений, часто реализуемых на практике [13]:

- функциональная СППР;
- независимые витрины данных;
- двухуровневое хранилище данных;
- трехуровневое хранилище данных.

2.2.1. Функциональная СППР

Функциональная СППР (рисунок 2.2) является наиболее простой с архитектурной точки зрения. Такие системы часто встречаются на практике, в особенности в организациях с невысоким уровнем аналитической культуры и недостаточно развитой информационной инфраструктурой.

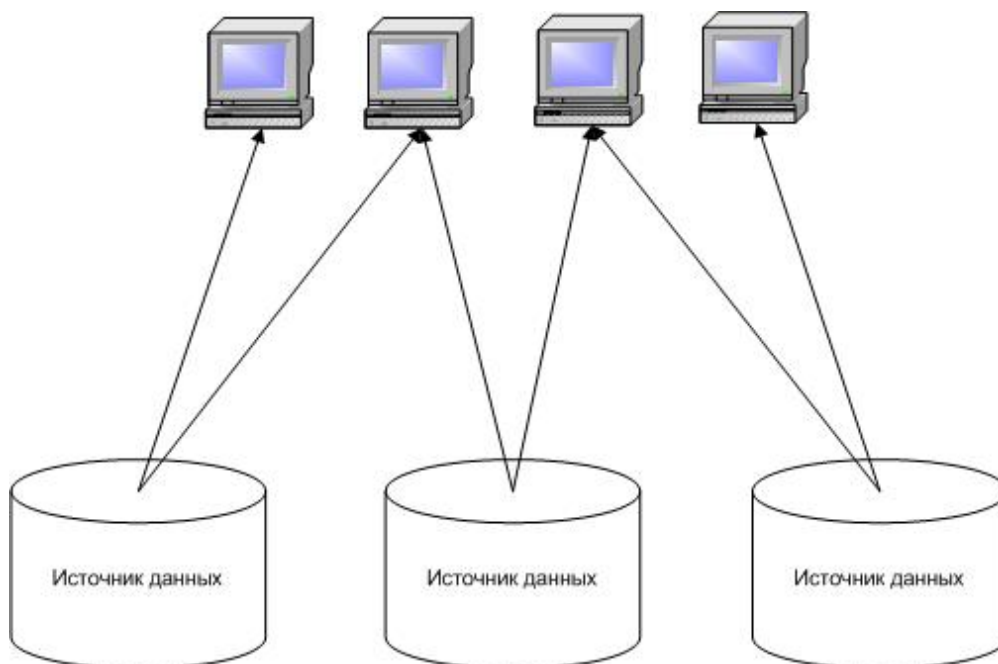


Рисунок 2.2. Функциональная система поддержки принятия решений

Характерной чертой функциональной СППР является то, что анализ осуществляется с использованием данных из оперативных систем.

Преимущества:

- быстрое внедрение за счет отсутствия этапа перегрузки данных в специализированную систему;
- минимальные затраты за счет использования одной платформы.

Недостатки:

- единственный источник данных, потенциально сужающий круг вопросов, на которые может ответить

система;

- оперативные системы характеризуются очень низким качеством данных с точки зрения их роли в поддержке принятия стратегических решений. В силу отсутствия этапа очистки данных, данные функциональной СППР, как правило, обладают невысоким качеством;
- большая нагрузка на оперативную систему. Сложные запросы могут привести к остановке работы оперативной системы, что весьма нежелательно.

2.2.2. СППР с использованием независимых витрин данных

В СППР с использованием независимых витрин данных (рисунок 2.3) в зависимости от предметной области анализа формируются витрины данных, куда поступает информация из внешних источников данных. Пользователи (аналитики) для решения конкретных вопросов работают только с необходимыми витринами данных.

Независимые витрины данных часто появляются в организации исторически и встречаются в крупных организациях с большим количеством независимых подразделений, зачастую имеющих свои собственные отделы информационных технологий.

Преимущества:

- витрины данных можно внедрять достаточно быстро;
- витрины проектируются для ответов на конкретный ряд вопросов;
- данные в витрине оптимизированы для использования

определенными группами пользователей, что облегчает процедуры их наполнения, а также способствует повышению производительности;

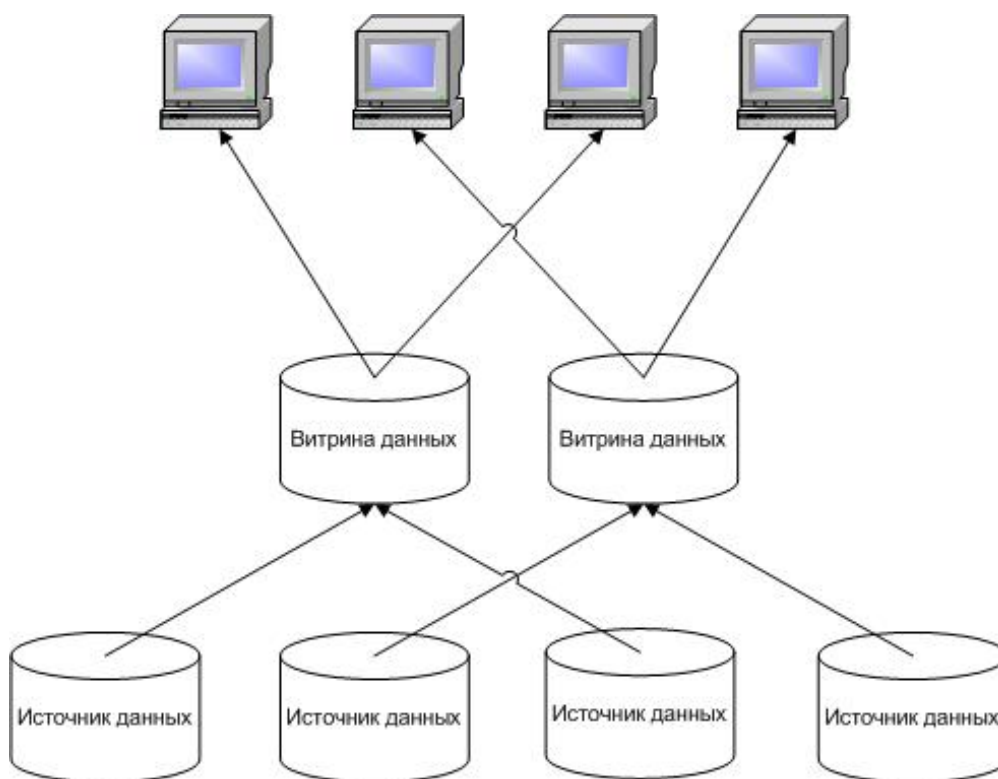


Рисунок 2.3. Независимые витрины данных

Недостатки:

- данные хранятся многократно в различных витринах данных. Это приводит к дублированию данных и, как следствие, к увеличению расходов на хранение и потенциальным проблемам, связанным с необходимостью поддержания непротиворечивости данных;
- потенциально очень сложный процесс наполнения витрин данных при большом количестве источников данных;
- данные не консолидируются на уровне предприятия, таким образом, отсутствует единая картина бизнеса.

2.2.3. СППР на основе двухуровневого хранилища данных

Двухуровневое хранилище данных (рисунок 2.4) строится централизованно для предоставления информации в рамках компании. Для поддержки такой архитектуры необходима выделенная команда профессионалов в области хранилищ данных.

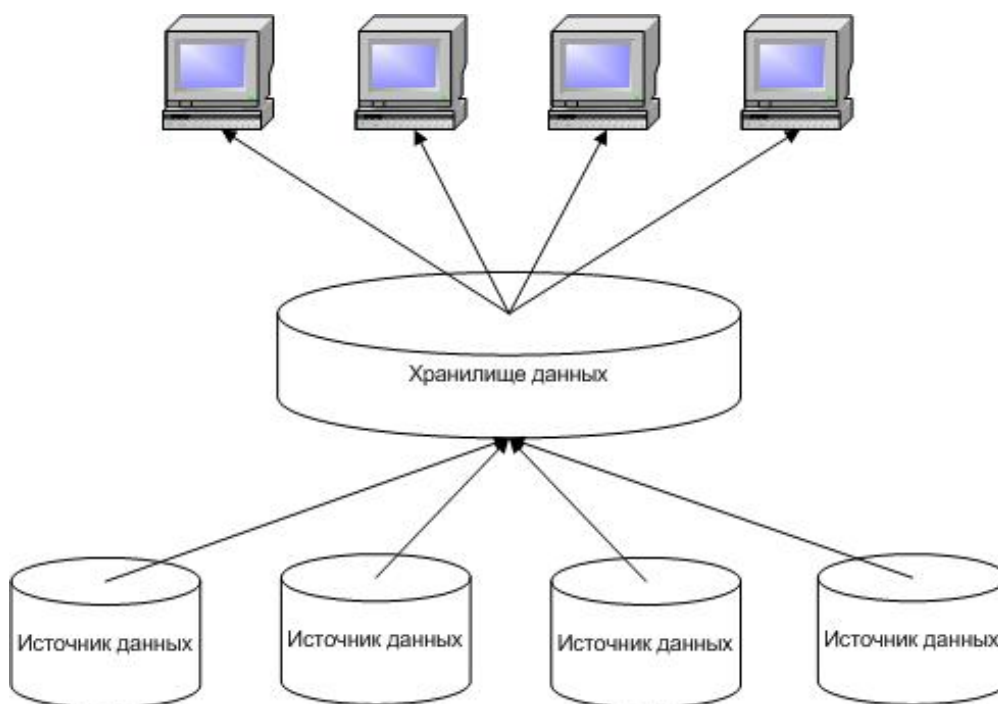


Рисунок 2.4. Двухуровневое хранилище данных

Это означает, что вся организация должна согласовать все определения и процессы преобразования данных.

Преимущества:

- данные хранятся в единственном экземпляре;
- минимальные затраты на хранение данных;
- отсутствуют проблемы, связанные с синхронизацией нескольких копий данных;
- данные консолидируются на уровне предприятия, что позволяет иметь единую картину бизнеса.

Недостатки:

- данные не структурируются для поддержки потребностей отдельных пользователей или групп пользователей;
- возможны проблемы с производительностью системы;
- возможны трудности с разграничением прав пользователей на доступ к данным.

2.2.4. СППР на основе трёхуровневого хранилища данных

Трёхуровневое хранилище данных представляет собой единый централизованный источник корпоративной информации (рисунок 2.5).

Витрины данных представляют подмножества данных из хранилища, организованные для решения задач отдельных подразделений компании. Конечные пользователи имеют возможность доступа к детальным данным хранилища, в случае если данных в витрине недостаточно, а также для получения более полной картины состояния бизнеса.

Преимущества:

- создание и наполнение витрин данных упрощено, поскольку наполнение происходит из единого стандартизованного надежного источника очищенных нормализованных данных;
- витрины данных синхронизированы и совместимы с корпоративным представлением. Имеется корпоративная модель данных. Существует возможность сравнительно лёгкого расширения

хранилища и добавления новых витрин данных;

- гарантированная производительность.

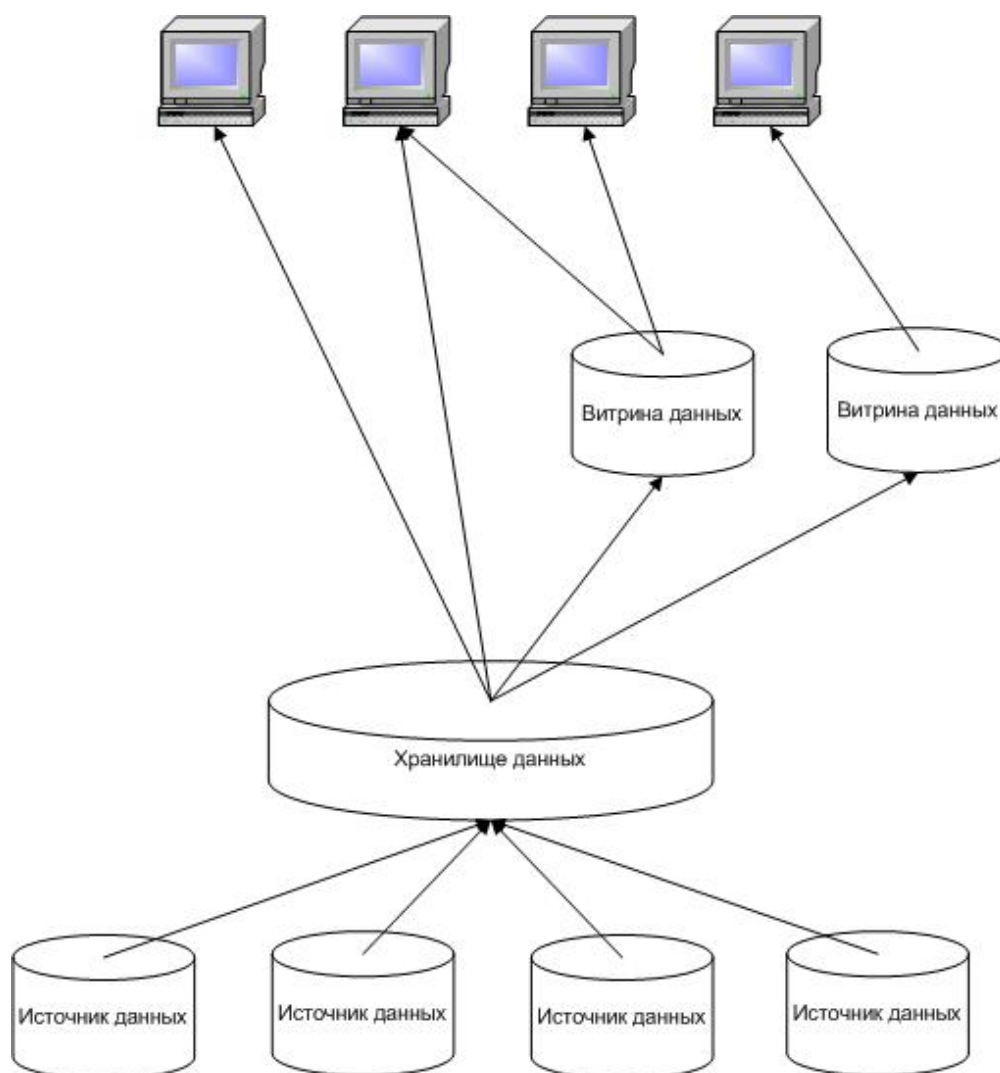


Рисунок 2.5. Трехуровневое хранилище данных

Недостатки:

- существует избыточность данных, ведущая к росту требований на хранение данных;
- требуется согласованность с принятой архитектурой многих областей с потенциально различными требованиями (например, скорость внедрения иногда конкурирует с требованиями следовать архитектурному подходу).

По результатам проведенного анализа базовой

архитектурой для разработки системы поддержки принятия решений и мониторинга проектов была выбрана функциональная архитектура СППР. В этом типе архитектуры основными недостатками являются низкое качество исходных для анализа данных с точки зрения их роли в поддержке принятия стратегических решений и повышенная нагрузка на системы оперативной обработки данных за счет выполнения сложных запросов, которые могут агрегировать информацию из множества различных таблиц. Проблему качества исходной информации можно решить за счет обработки и преобразования извлеченной информации на сервере приложений перед передачей данных в подсистемы анализа информации. К сожалению, данное решение увеличивает нагрузку на сервер приложений и суммарное время обработки информации.

Использование достаточно агрессивного кэширования не только извлеченной и преобразованной информации, но результатов последующего анализа данных позволяет снизить нагрузку как на OLTP системы источников данных, так и сервер приложений. В действительности для принятия стратегических решений руководителю важно представлять ситуацию в целом, в то время как некоторые отклонения за счет не своевременной актуализации информации не существенно влияют на картину в целом. Подобрав оптимальные параметры для размера, времени истечения хранения информации в оперативной памяти по типам используемой информации и других параметров кэширования, можно существенно снизить

нагрузку на сервер приложений и серверы баз данных.

Тем не менее преимущества функциональной архитектуры СППР, такие как быстрота внедрения за счет отсутствия этапа перегрузки данных в специализированную систему и минимализация затрат за счет использования единой платформы, существенно повлияли на выбор данного типа архитектурной организации системы.

2.3. Методы интеллектуального анализа данных (Data Mining)

Существует несколько подходов к классификации методов интеллектуального анализа данных. Рассмотрим схему классификации, основанной на разделении подходов по отношению к обучению математических моделей. Таким образом многообразие методов Data Mining разделяют на две группы [14]:

- статистические методы, основанные на использовании усредненного накопленного опыта, который отражен в ретроспективных данных;
- кибернетические методы, включающие множество разнородных математических подходов.

Недостаток такой классификации: и статистические, и кибернетические алгоритмы тем или иным образом опираются на сопоставление статистического опыта с результатами мониторинга текущей ситуации.

Преимуществом такой классификации является ее удобство для интерпретации — она используется при описании математических средств современного подхода к извлечению

знаний из массивов исходных наблюдений (оперативных и ретроспективных), т.е. в задачах Data Mining.

Рассмотрим подробнее представленные выше группы.

Статистические методы Data mining

Данные методы представляют собой четыре взаимосвязанных раздела:

- предварительный анализ природы статистических данных (проверка гипотез стационарности, нормальности, независимости, однородности, оценка вида функции распределения, ее параметров и т.п.);
- выявление связей и закономерностей (линейный и нелинейный регрессионный анализ, корреляционный анализ и др.);
- многомерный статистический анализ (линейный и нелинейный дискриминантный анализ, кластерный анализ, компонентный анализ, факторный анализ и др.);
- динамические модели и прогноз на основе временных рядов.

Статистические методы Data Mining классифицированы на четыре группы методов:

- дескриптивный анализ и описание исходных данных;
- анализ связей (корреляционный и регрессионный анализ, факторный анализ, дисперсионный анализ);
- многомерный статистический анализ (компонентный анализ, дискриминантный анализ, многомерный регрессионный анализ, канонические корреляции и

др.);

- анализ временных рядов (динамические модели и прогнозирование).

Кибернетические методы Data Mining

Второе направление Data Mining — это множество подходов, объединенных идеями компьютерной математики и использования теории искусственного интеллекта.

К этой группе относятся такие методы:

- искусственные нейронные сети (распознавание, кластеризация, прогноз);
- эволюционное программирование (в т.ч. алгоритмы метода группового учета аргументов);
- генетические алгоритмы (оптимизация);
- ассоциативная память (поиск аналогов, прототипов);
- нечеткая логика;
- деревья решений;
- системы обработки экспертных знаний.

2.4. Архитектура кроссплатформенной системы поддержки принятия решений

Архитектура кроссплатформенной системы поддержки принятия решений и объектного мониторинга представляет собой классическое трехзвенное приложение (рисунок 2.6).

Консолидация данных происходит из различных независимых источников данных, которыми являются базы данных оперативных систем управления проектами различных ведомств, на сервере приложений (Application Server). Представление данных реализуется посредством

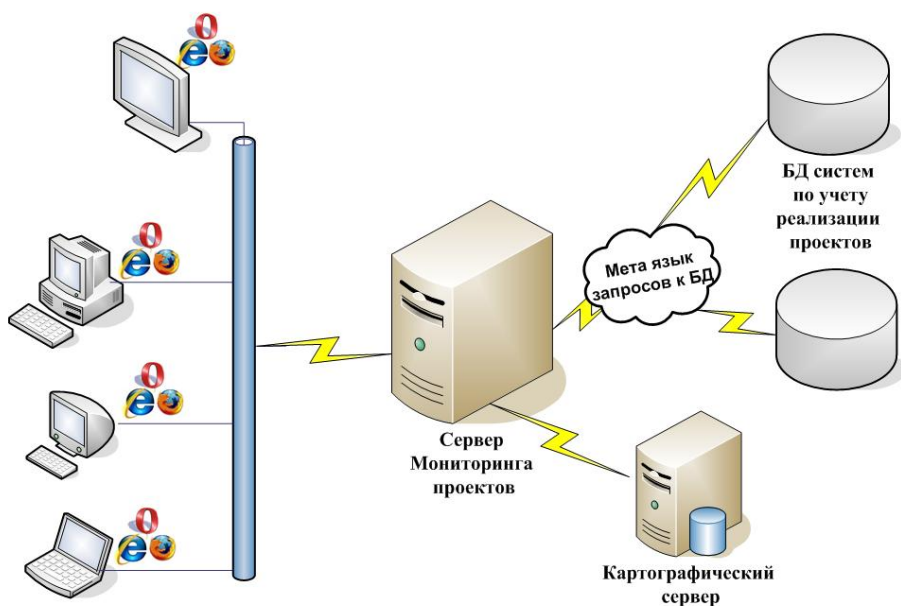


Рисунок 2.6. Архитектура системы мониторинга проектов отображения на клиенте в веб-браузере. Использование тонкого клиента обусловлено необходимостью поддержки кроссплатформенности клиентской части автоматизированной системы.

Сервер приложений включает в себя несколько подсистем:

- загрузки информации из источников;
- обработки и преобразования загруженной информации;
- агрегации и формирования отчетности;
- интеграции проектной информации и данных с картографического сервера;
- взаимодействия с пользователями.

Схема, описывающая структуру информационной системы, представлена на рисунке 2.7. С помощью менеджера извлечения данных происходит загрузка информации из источников данных и гео-информационных систем в соответствии с определенными критериями отбора данных.



Рисунок 2.7. Структура системы поддержки принятия решений и мониторинга проектов

В действительности у всех типов проектов, с понятием которого в качестве ключевой сущности оперирует разработанная система поддержки принятия решений, можно выделить основные характеристики:

- название;
- местоположение (пространственные координаты);
- ответственные ведомства;
- ответственные министерства;
- сроки реализации;
- бюджет;
- степень готовности и др.

В зависимости от принимаемого руководителем решения и необходимой для этого информации формируются разнообразные критерии отбора данных, которые могут

быть как заложены изначально в систему, так и иметь динамический характер, позволяя пользователю самому определять параметры и их значения для формирования запросов. Последнее требует от пользователя более высокой подготовки, по сравнению с вариантом заранее определенных критериев, и зачастую требует предварительного обучения для работы с информационной системой.

Менеджер загрузки производит обработку, преобразование и унификацию данных для дальнейшего использования другими подсистемами. Следует заметить, что для хранения информации не используется внутренняя база данных с выгрузкой информации в нее из источников. Поэтому на менеджере загрузки лежит ответственность не только за загрузку информации, но и ее предварительную обработку, выявлению и исправлению ошибок. Как уже говорилось, для уменьшения нагрузки на сервер приложений и базы данных в приложении активно применяется кэширование данных, что также инкапсулировано в подсистеме загрузки данных.

В своей работе менеджер загрузки получает запросы на предоставление данных от других подсистем приложения (подсистемы анализа и формирования отчетности, представления данных, интеграции с картографическим сервером), анализирует текущее состояние загруженных в память объектов (в случае их отсутствия или истечения времени хранения в оперативной памяти происходит обращение к базе данных источника) и формирует ответ на запрос.

На следующем этапе происходит анализ информации, агрегирование и подсчет статистики, прогнозирование и подготовка отчетности. Затем сводная информация отображается пользователю в текстовом формате, в виде географической карты или сводных графиков.

2.5. Описание алгоритма работы менеджера загрузки

Подсистема загрузки информации является ключевой подсистемой информационной системы, в которой заложены не только алгоритмы обработки, преобразования и унификации данных для дальнейшего использования их другими подсистемами, но и методы предварительной обработки, выявления и исправления ошибок.

Диаграмма, описывающая алгоритм извлечения информации из источников данных при поступлении запроса к подсистеме загрузки информации, представлена на рисунке 2.8.

Модуль загрузки информации состоит из двух подсистем:

1. подсистем непосредственного извлечения данных из OLTP систем источников данных («Подсистема извлечения данных из источника данных»);
2. подсистемы очистки, преобразования и интеграции извлеченной информации («Менеджер загрузки»).

Подсистемы извлечения данных из OLTP систем выполняют функции формирования SQL запросов к источникам данных, их выполнения и формирования объектов классов на основе полученных результатов, извлечения и помещения объектов в кэш и инкапсуляции алгоритмы агрегации и конвертации извлеченных данных.

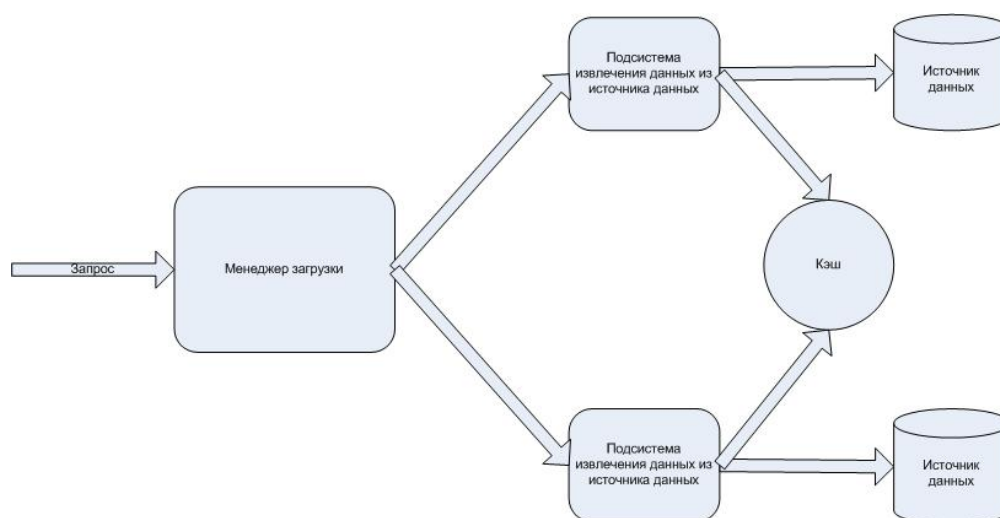


Рисунок 2.8. Алгоритм работы менеджера загрузки

Менеджер загрузки является звеном в архитектуре системы, который скрывает реализацию процессов извлечения информации из конкретных источников данных от других подсистем информационной системы и выполняет функции агрегации и подготовки данных к передаче инициировавшей запрос подсистеме.

При поступлении запроса на извлечение информации к менеджеру загрузки в виде вызова метода происходит анализ конфигурационного файла на предмет выявления подключенных источников данных, из которых может быть загружена информация, определяются источники данных, к которым необходимо перенаправить запрос, и вызываются соответствующие методы подсистем извлечения данных из OLTP систем.

Далее происходит проверка кэш-памяти на наличие результатов запроса в оперативной памяти. В случае их присутствия в кэше происходит их извлечение из оперативной памяти. В противном случае формируются и исполняются SQL запросы к базам данных источников информации и результаты

запросов помещаются в кэш-память. Полученные результаты преобразуются в объекты классов и возвращаются в виде ответа менеджеру загрузки.

Очевидно, что описанный этап загрузки данных хорошо поддается распараллеливанию в плане выполнения работы подсистем извлечения данных из OLTP систем в виде отдельных потоков. С одной стороны данный подход позволяет увеличить временную эффективность работы алгоритма и уменьшить суммарное время извлечения информации из источников. С другой стороны использование многопоточности увеличивает сложность реализации алгоритма и может приводить к трудно обнаруживаемым ошибкам, которые достаточно сложно воспроизводятся в тестовых условиях и, как следствие, плохо диагностируются.

Менеджер загрузки производит слияние полученной информации. Следующим этапом является приведение полученной разнотипной информации к унифицированным типам и классам, с которыми будут работать другие подсистемы разрабатываемой информационной системы. На данном этапе происходит вызов классов-конверторов, которые производят преобразование и очистку исходной информации. Стоит заметить, что алгоритмы конвертеров являются уникальными для каждого источника данных.

Цикл работы модуля загрузки информации заканчивается на том, что преобразованная информация возвращается подсистеме, инициировавшей запрос.

ГЛАВА 3. ОПИСАНИЕ КРОССПЛАТФОРМЕННОЙ СИСТЕМЫ ПОДДЕРЖКИ ПРИНИЯТИЯ РЕШЕНИЙ И МОНИТОРИНГА ПРОЕКТОВ

3.1. Диаграмма классов предметной области

Модель данных предметной области системы поддержки принятия решений и мониторинга проектов представлена следующими сущностями:

- сотрудник;
- министерство;
- детали проекта;
- проект;
- муниципальное образование;
- объект проекта;
- фотография.

Связи между сущностями в рассматриваемой модели данных информационной системы представлены на рисунке 3.1.

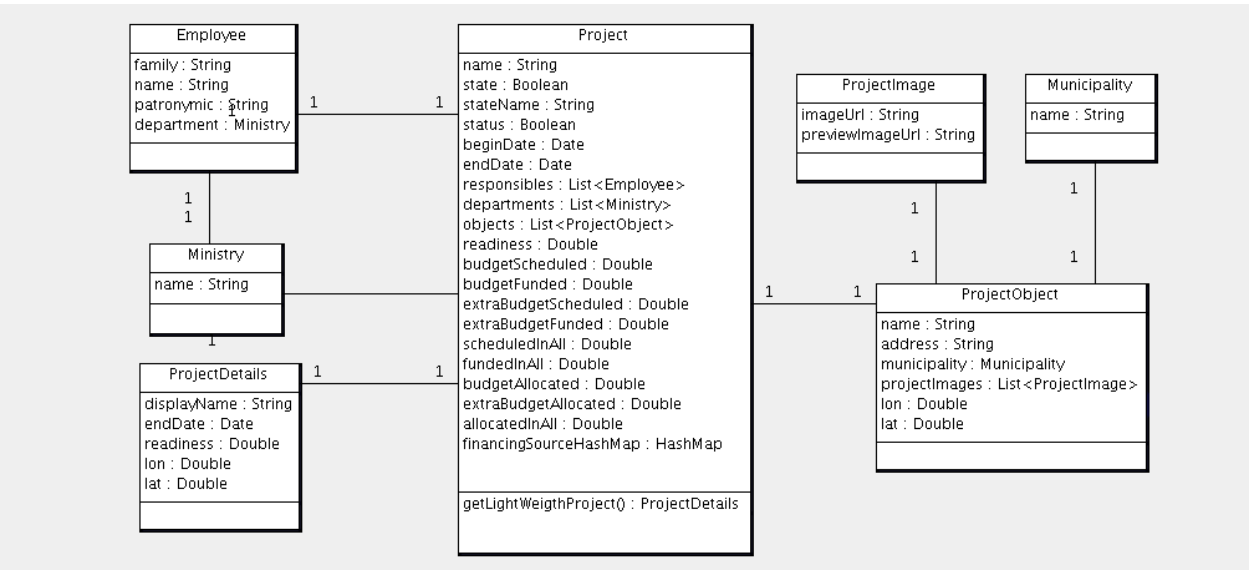


Рисунок 3.1. Диаграмма классов предметной области

«Проект» является ключевой сущностью предметной области разработанной информационной системы и представлен следующими характеристиками:

- название;
- состояние;
- статус;
- дата начала работ по проекту;
- планируемая дата окончания работ;
- ответственные лица;
- курирующие ведомства;
- список объектов, на которых ведутся работы в рамках проекта;
- степень готовности;
- источники финансирования;
- количество бюджетных средств, запланированных к выделению;
- количество профинансированных бюджетных средств;
- количество выделенных бюджетных средств;
- количество внебюджетных средств, запланированных к выделению;
- количество профинансированных внебюджетных средств;
- количество выделенных внебюджетных средств.

Сущность «Проект» обладает достаточно широким набором параметров. Но не всегда целесообразно передавать всю информацию о рассматриваемом проекте на клиент с сервера. В связи с этим введена дополнительная сущность

— «облегченная» версия «Проекта», в которой отсутствуют большинство детализированных параметров. «Облегченная» версия включает в себя:

- название;
- дату окончания работ по проекту;
- степень готовности;
- географические координаты проекта.

В действительности проект может включать в себя проведение работ на нескольких распределенных в пространстве и независимых физических объектах. Сущность «Объект проекта» описывает объекты, на которых ведутся работы в рамках проекта. «Объект проекта» включает в себя следующие характеристики:

- название объекта;
- адрес;
- муниципальное образование;
- фотоматериалы;
- географические координаты.

Одной из характеристик описания объекта проекта является муниципальное образование. Сущность «Муниципальное образование» по сути является пространственной характеристикой объекта, т.е. с помощью данного свойства определяется положение объекта на географической карте. Тем не менее объект может иметь более определенное местоположение внутри муниципального образования, если такая информация доступна в источнике данных.

Очевидно, что ответственными за реализацию этапов проекта являются министерства или подразделения учреждения, ведущего проект. В связи с этим определена сущность «Министерство», которая характеризуется единственной характеристикой — названием министерства. Действительно, для анализа информации об учете ведения проектов иная информация о министерствах не представляет для нас интереса.

Также в доменной модели предметной области выделены такие сущности, как «Сотрудник» и «Фотография». Детально их характеристики представлены выше (см. рисунок 3.1).

3.2. Архитектура приложения и инструменты разработки

В основе архитектуры приложения лежит классическое решение — шаблон проектирования Model-View-Presenter (MVP) [16]. В действительности MVP считается производным от паттерна проектирования Model-View-Controller (MVC) [41]. Основным отличием является отсутствие возможности оповещения модели о своем изменении представления. Данное допущение приводит к упрощению реализации архитектуры приложения. Основным достоинством данного подхода является разделение бизнес-логики приложения от представления данных, что приводит к облегчению процесса автоматического тестирования и уменьшению связей между модулями системы.

View определяется как интерфейс, который Presenter использует для получения и установки данных модели.

Использование интерфейса вместо конкретных реализаций классов уменьшает связанность между модулями и позволяет производить безболезненную подмену представлений для отображения данных. Реализация Presenter содержит ссылку на интерфейс View. Параметром конструктора Presenter является интерфейс View, в то время, как в конструктор передается конкретная реализация интерфейса View. Когда в представлении (View) возникает событие, вызывается конкретный метод Presenter. Затем Presenter получает данные из представление посредством представления View. Далее Presenter вызывает методы модели, получает результаты и устанавливает данные из модели во View опять же через интерфейс [42].

С точки зрения многоуровневой модели Presenter может быть отнесен как к уровню приложения, так и к промежуточному уровню между уровнем приложения и уровнем пользовательского интерфейса. На мой взгляд, логичнее относить Presenter к промежуточному уровню: на Presenter возлагаются обязанности только по обработке пользовательских событий и получению/установке данных из представления. Бизнес-логика же представлена в модели [43] [44].

Таким образом, использование данного подхода при организации архитектуры приложения на верхнем уровне позволило:

- облегчить процесс автоматического тестирования;
- уменьшить связанность между уровнями приложения;

- получить возможность изменения представления (View) без изменения других уровней приложения при необходимости;
- отделить бизнес-логику от представления данных;
- воспользоваться проверенным архитектурным решением при разработке системы.

Более пристальное внимание хотелось бы уделить архитектурному решению при реализации подсистемы загрузки информации из источников данных (рисунок 3.2).

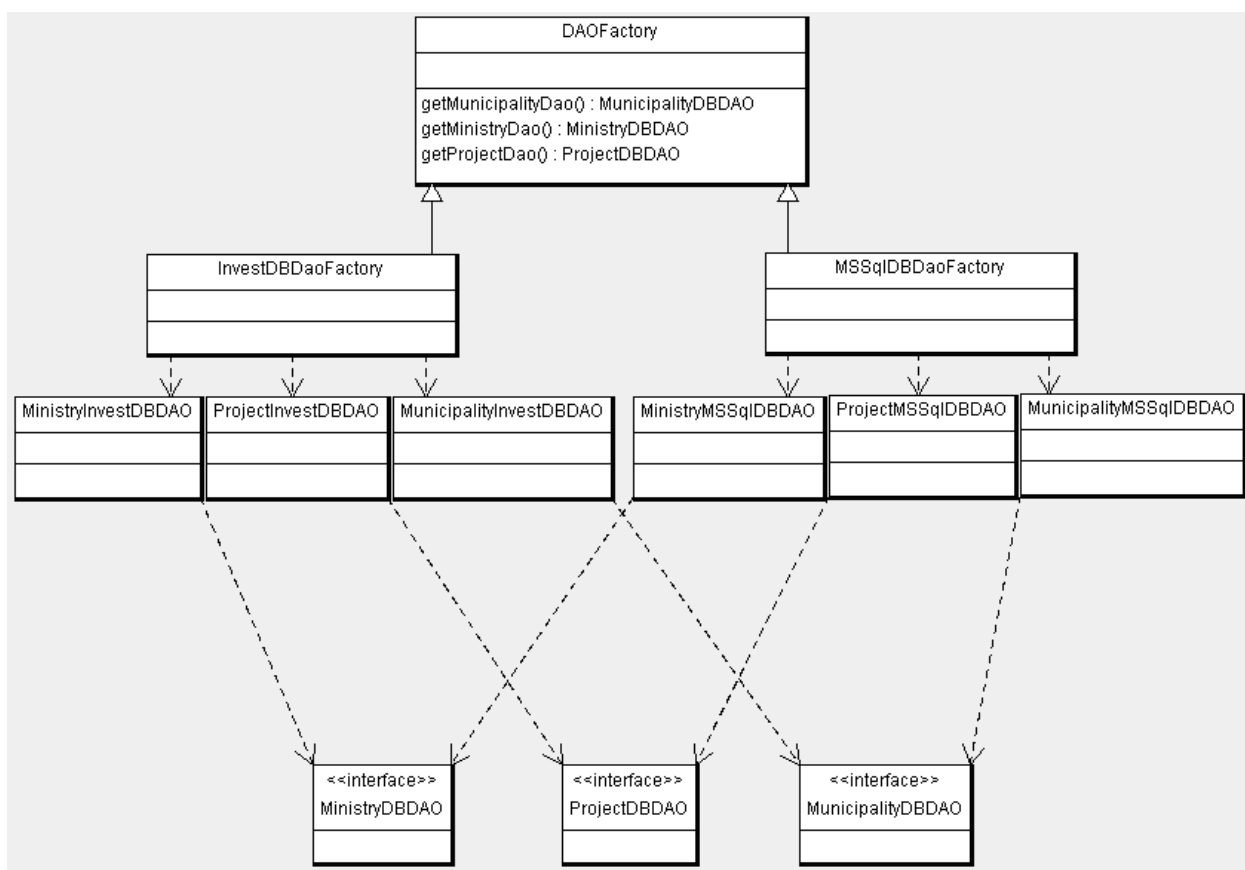


Рисунок 3.2. Архитектуру подсистемы загрузки информации

Важным требованием к данной подсистеме является возможность подключения новых источников данных без существенного изменения существующего программного кода. В связи с этим была реализована двухуровневая архитектурная

модель: классы первого уровня являются открытыми для взаимодействия с другими подсистемами в то время, как классы второго уровня инкапсулированы в подсистеме загрузки информации.

Первый уровень представлен интерфейсами MinistryDAO, MunicipalityDAO и ProjectDAO. Для других подсистем приложения данные интерфейсы являются типичными DAO (Data Access Object), которые представляют возможность для взаимодействия с данными, находящимися в хранилище данных (база данных, xml, файл и др.). В действительности DAO классы первого уровня не работают напрямую с источниками данных, а всего лишь перенаправляют запросы к конкретным реализациям DAO для соответствующих запросу источникам данных, проводят преобразование полученных данных к модели данных, описанной выше, и организуют слияние преобразованных данных.

Второй архитектурный уровень содержит в себе реализации DAO классов для конкретных источников данных. Их реализация достаточно типична для задачи извлечения данных и основана на принципе CRUD (Create Read Update Delete). Извлечение данных и преобразование к объектной модели происходит посредством Java Persistence API 2.0.

Серверная часть проекта была написана на объектно-ориентированном языке программирования Java 6 [15]. Для организации доступа к данным использовался Java Persistence API 2.0 (JPA 2.0). В качестве JPA провайдера выступил Hibernate. Кэширование данных обеспечивается с помощью

Ehcache.

Для разработки клиентской части приложения использовался фреймворк Google Web Toolkit (GWT), который позволяет создавать Ajax приложения на основе Java.

3.3. Экспериментальная проверка алгоритмов

Для оценки корректности функционирования основных модулей информационной системы применялись как методы автоматического тестирования программного кода, так и ручная прогонка заранее подготовленных тестов. Для автоматической проверки корректности реализованных алгоритмов использовалась библиотека тестирования JUnit версии 4.5.

Методология модульного тестирования, которой следует JUnit, позволяет начинать тестировать программный код по сути ещё до того момента, когда код написан. При использовании методологии TDD (Test Driven Development) состояние программного кода находится в одной из двух фаз: все тесты успешно пройдены или на каком-то из тестов программа отработала не верно. Это приводит к следующему циклу разработки. Изначально пишется простой тест, который не проходит, т.к. ещё не написан программный код, который тестируется. Далее пишется версия метода, которая проходит тест, и на данном тесте программа отработывает верно. Затем пишется ещё тест и метод теперь тестируется уже на двух тестах. Метод модифицируется таким образом, чтобы новый тест снова проходил. Процесс повторяется до тех пор, пока количество тестов и покрытие возможных сочетаний входных

параметров не станет приемлемым.

Стоит заметить, что разрабатываемые тесты должны тестировать минимальный структурный блок программы — метод. Ко всему прочему тесты должны выполняться достаточно быстро, чтобы не вызывать существенные задержки в работе программиста в плане ожидания результатов прогонки всех модульных тестов. Программист должен регулярно запускать тесты, что быть уверенным, что внесенные изменения в программный код не вызвали неожиданного поведения в других подсистемах и модулях программы. Отсюда логично предположить, что не желательно тестирование модульными тестами методов, которые работают с внешними ресурсами (базами данных, файлами, сетью), т.е. методов, выполнение которых занимает значительное время, но не мгновенно.

Классы, предназначенные для загрузки данных из источников данных, были протестированы следующим образом. В тестовые источники данных, используемые для отладки системы, был занесен заранее определенный набор данных. Например, в первый источник данных была записана информация о пяти проектах, а во второй — о других трех. Далее запускались методы соответствующих классов и результаты их работы сравнивались с ожидаемыми. В принципе подобный процесс можно автоматизировать с использованием методологии модульного тестирования, но в таком случае мы теряем главное их преимущество — скорость прогона всего тестового набора.

3.4. Функции системы

Разработанная информационная система поддержки принятия решений и мониторинга проектов включает в себя следующие функции:

- просмотр списка проектов, извлеченных из всех источников данных;
- фильтрация проектов по принадлежности к министерству, ведомству, муниципальному образованию;
- просмотр галереи фотографий, относящихся к проекту;
- просмотр информации о проекте;
- визуальное представление распределения проектов на карте Астраханской области.
- генерация статистической информации:
 - распределение количества проектов по ведомствам;
 - распределение количества проектов по муниципальным образованиям;
 - распределение финансирования по источникам финансирования;
 - финансирование по ведомствам;
 - отклонения от сроков выполнения проектов;
 - проекты, сроком окончания которых является текущий месяц.

Рассмотрим пользовательские интерфейсы информационной системы. Интерфейс с фильтрацией проектов по отношению к конкретному министерству или ведомству представлен на рисунке 3.3.

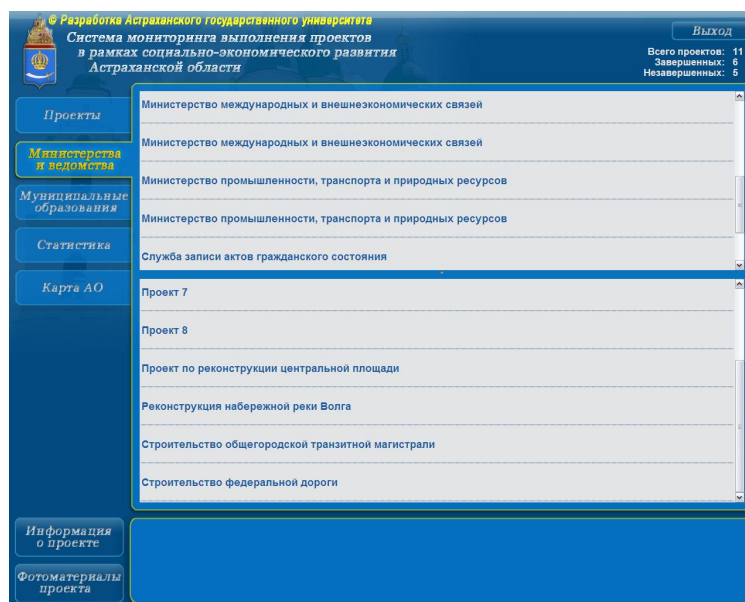


Рисунок 3.3. Интерфейс фильтрации проектов по отношению к министерству или ведомству

После выбора того или иного ведомства в верхней части экрана пользователю будут представлены все проекты, которые курируются данным ведомством. При выборе проекта в списке в нижней части экрана происходит загрузка фотоматериалов доступных по объектам проекта. Также можно переместиться в фотогалерею и просмотреть все доступные фотографии, касающихся рассматриваемого проекта, или переститься на интерфейс демонстрации информации о проекте, щелкнув на соответствующую кнопку («Информация о проекте») в левой нижней части экрана.

Фильтрация проектов по муниципальным образованиям, которая организована по тому же принципу, что и фильтрация по ведомствам, представлена на рисунке 3.4.

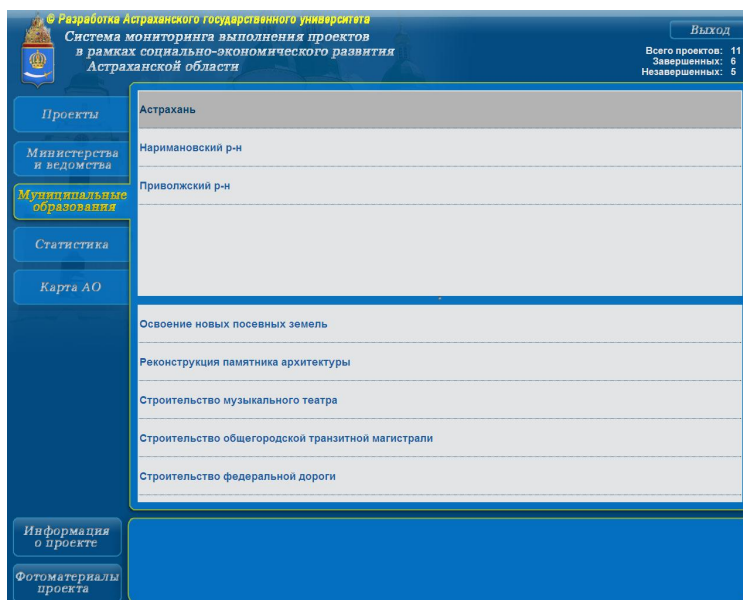


Рисунок 3.4. Интерфейс фильтрации проектов по отношению к муниципальному образованию

Далее мы переместились уже на вкладку «Статистика» (рисунок 3.5). На ней представлена возможность выбора разнотипных вариантов получения отчетной информации. В частности на рисунке 3.5 представлено распределение финансирования по конкретным ведомствам. Мы можем наблюдать сколько денег запланировано к выделению, выделено и уже профинансировано на различные проекты, относящиеся к конкретному ведомству.



Рисунок 3.5. Распределение финансирования по ведомствам

Интерфейс с картой Астраханской области с нанесенными на нее метками-проектами представлен на рисунке 3.6. Щелчок мышкой по метке перенаправит нас на интерфейс демонстрации информации о проекте.

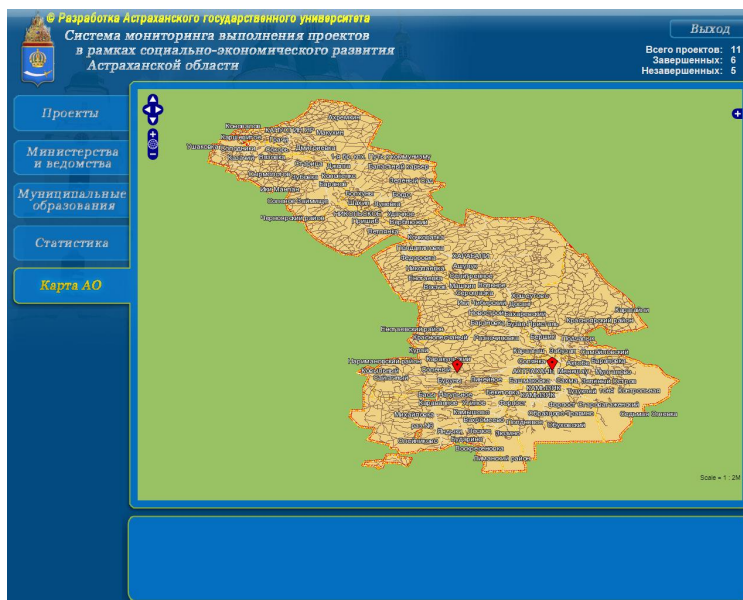


Рисунок 3.6. Карта Астраханской области с нанесенными метками-проектами

Интерфейс демонстрации информации о проекте демонстрирует характеристики рассматриваемого проекта (рисунок 3.7).

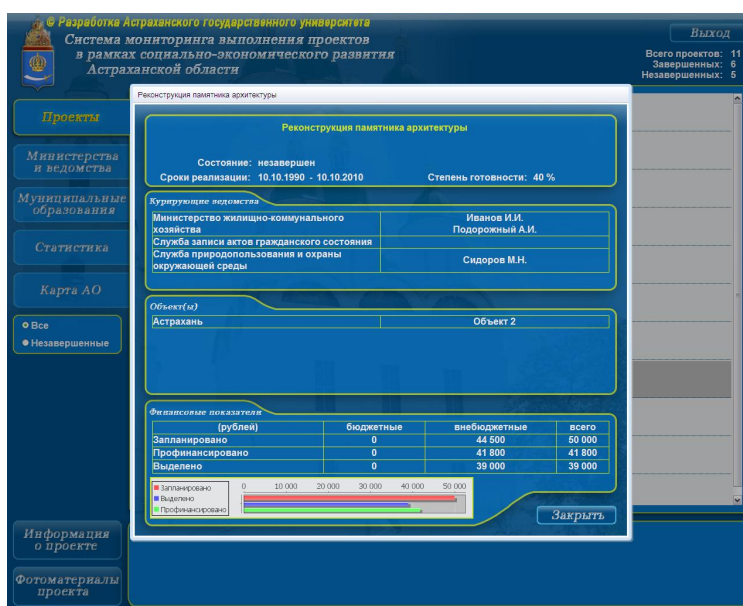


Рисунок 3.7. Информация о проекте

Интерфейс представляет следующие характеристики рассматриваемого проекта

- название проекта;
- состояние;
- сроки реализации;
- степень готовности;
- курирующие министерства и ведомства;
- относящиеся к проекту объекты;
- обобщенные финансовые показатели;
- диаграмма распределения типов финансирования.

3.5. Перспективы развития

Несмотря на то, что задачи, поставленные перед разработкой описанной информационной системы, решены, в текущей версии программного продукта присутствуют следующие недостатки:

- повышенная нагрузка на сервер приложений и базы данных источников информации в момент извлечения информации в связи с отсутствием специализированного хранилища данных, в которое выгружаются данные из OLTP систем. В представленной системе нагрузку удастся снизить за счет подбора параметров для агрессивного кэширования наиболее используемых данных;
- отсутствие возможности подключения нового источника данных с помощью графических конструкторов без написания отдельного программного модуля к системе.

В действительности отсутствие возможности подключения нового источника данных с помощью

графических конструкторов трудно назвать недостатком. Скорее создание подобных инструментов является логичным продолжением развития приложения и механизмом упрощения интеграционных работ при настройке системы.

Что касается проблем с нагрузкой на сервер приложений, то тут совершенно нет однозначного и полностью удовлетворительного решения на данный момент. Одним из вариантов является создание промежуточного хранилища с синхронизацией данных с источниками данных и хранением результатов запросов.

ЗАКЛЮЧЕНИЕ

В рамках магистерской диссертации для достижения поставленной цели были проанализированы существующие автоматизированные системы поддержки принятия решений и мониторинга проектов и сформулированы требования, которым должна удовлетворять система; спроектирована архитектура системы; разработаны механизмы взаимодействия с внешними разнородными информационными системами; проведено исследование существующих математических методов анализа фактографической информации; выбраны необходимые средства разработки; создана кроссплатформенная информационно-аналитическая система управления проектами.

Научная новизна результатов, полученных в диссертации, заключается в следующем: создана модель кроссплатформенной системы, отличающейся возможностью консолидации информации из разнородных источников данных о проектной деятельности; разработан оригинальный алгоритм интеграции информации из разнородных баз данных.

По результатам теоретических исследований была разработана кроссплатформенная система поддержки принятия решений и мониторинга проектов, позволяющая проводить контроль выполнения проектных работ, получать актуальную и своевременную информацию по реализации проектов различными ведомствами, представлять пользователю аналитическую информацию.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. И.Н. Горбачев, Р.Ж. Кушалиев, А.А. Павлов. О.В. Щербинина. «Кроссплатформенная система поддержки принятия решений и мониторинга проектов». Инновационные технологии в управлении, образовании, промышленности «АСТИНТЕХ-2010» [Текст]: материалы Международной научной конференции 11–14 мая 2010 г.: в 3 т. / сост. И.Ю. Петрова. — Астрахань: Астраханский государственный университет, Издательский дом «Астраханский университет», 2010. — Т.1. — 206 с.
2. И. Петрова, М. Зайцева, О. Щербинина, Е. Эрман, Н. Тюрин «Автоматизированная система принятия решений и объектного мониторинга мероприятий, проводимых на территории региона». Proceedings of the International Conference «e-Management & Business Intelligence», Varna, 2007. Sofia, Institute of Information Theories and Applications FOI ITHEA-2007.
3. Microsoft Project. [Электронный ресурс]. // Официальный сайт Wikipedia. — Электронные данные. — Режим доступа: свободный, http://ru.wikipedia.org/wiki/Microsoft_Project. — Заглавие с экрана. — Яз. русский.
4. ProjectMate. [Электронный ресурс]. // Официальный сайт Wikipedia. — Электронные данные. — Режим доступа: свободный, <http://ru.wikipedia.org/wiki/ProjectMate>. — Заглавие с экрана. — Яз.

русский.

5. Возможности ProjectMate. [Электронный ресурс]. // Официальный сайт проекта ProjectMate. — Электронные данные. — Режим доступа: свободный, <http://www.projectmate.ru/product/index.html>. — Заглавие с экрана. — Яз. русский.
6. Core J2EE Patterns - Data Access Object. [Электронный ресурс]. // Официальный сайт J2EE. — Электронные данные. — Режим доступа: свободный, <http://java.sun.com/blueprints/corej2eepatterns/Patterns/DataAccessObject.html>. — Заглавие с экрана. — Яз. английский.
7. Data Access Object. [Электронный ресурс]. // Официальный сайт Wikipedia. — Электронные данные. — Режим доступа: свободный, http://ru.wikipedia.org/wiki/Data_Access_Object. — Заглавие с экрана. — Яз. русский.
8. З. Гамма, Р. Хелм, Р. Джонсон, Дж. Влиссидес Приемы объектно-ориентированного проектирования. Паттерны проектирования. — Спб: Питер, 2001. — 368 с.: ил. (Серия «Библиотека программиста»)
9. Decision support system. [Электронный ресурс]. // Официальный сайт Wikipedia. — Электронные данные. — Режим доступа: свободный, http://en.wikipedia.org/wiki/Decision_support_system. — Заглавие с экрана. — Яз. русский.
10. Power D.J. A Brief History of Decision Support Systems.

- [Электронный ресурс]. DSSResources.COM, World Wide Web, — Электронные данные. — Режим доступа: свободный, <http://DSSResources.COM/history/dsshistory.html>, — May 31, 2003. — Заглавие с экрана. — Яз. английский.
11. Power D. J. «What is a DSS?» The On-Line Executive Journal for Data-Intensive Decision Support, 1997. — v. 1. — N3.
 12. Архипенков С. Аналитические системы на базе Oracle Express OLAP. Проектирование, создание, сопровождение. — М.: ДИАЛОГ-МИФИ, 2000 — 320 с.
 13. К. Лисянский Архитектурные решения и моделирование хранилищ и витрин данных. [Электронный ресурс]. // Настольный журнал ИТ-руководителя «Директор», — Электронные данные. — Режим доступа свободный, <http://www.osp.ru/cio/2002/03/172076/>, — Заглавие с экрана. — Яз. русский.
 14. И.А. Чубукова Методы и стадии Data Mining. [Электронный ресурс]. // Интернет университет информационных технологий, — Электронные данные. — Режим доступа свободный, <http://www.intuit.ru/department/database/datamining/3/2.html>, — Заглавие с экрана. — Яз. русский.
 15. Ноутон П., Шилдт Г. Java 2: Пер. с англ. — Спб.: БХВ-Петербург, 2001. — 1072с.: ил.
 16. Гамма Э., Хелм Р., Джонсон Р., Влиссидес Дж. Приемы объекто-ориентированного проектирования. Паттерны

- проектирования. — СПб: Питер, 2009. — 366 с.: ил.
17. Г. Буч Объектно-ориентированный анализ и проектирование с примерами приложений на C++, 2-е изд./Пер. с англ. — М.: «Издательство Бином», СПб.: «Невский диалект», 2001. — 560 с., ил.
 18. Мартин Р. Чистый код: создание, анализ и рефакторинг. Библиотека программиста. — СПб.: Питер, 2011. — 464 с.: ил.
 19. Спольски Дж. Джоэл о программировании. — Пер. с англ. — СПб.: Символ-Плюс, 2008. — 352 с., ил.
 20. Дж. Кью, М. Джеанини Объектно-ориентированное программирование. Учебный курс. — СПб.: Питер, 2005. — 238 с.: ил.
 21. Кингсли-Хью Э., Кингсли-Хью К. JavaScript 1.5: учебный курс. — СПб.: Питер, 2001. — 272 с.: ил.
 22. Крачтен, Филипп. Введение в Rational Unified Process. 2-е изд.: Пер. с англ. — М.: Издательский дом «Вильямс», 2002. — 240 с.: ил. — Парал. тит. англ.
 23. Лавров С.С. Программирование. Математические основы, средства, теория. — СПб.: БХВ-Петербург, 2001. — 320 с.: ил.
 24. Sumit Chandel. Using GWT with Hibernate. [Электронный ресурс]. //GWT Tutorials, — Электронные данные. — Режим доступа свободный, http://code.google.com/webtoolkit/articles/using_gwt_with_hibernate.html, — Заглавие с экрана. — Яз. английский.

25. Sumit Chandel. Testing Methodologies Using Google Web Toolkit. [Электронный ресурс]. //GWT Tutorials, — Электронные данные. — Режим доступа свободный, http://code.google.com/webtoolkit/articles/testing_methodologies_using_gwt.html, — Заглавие с экрана. — Яз. английский.
26. Daniel Wellman. Google Web Toolkit: Writing Ajax Applications Test First. Better Software magazine (November 2008).
27. Joel Webber. DOM Events, Memory Leaks, and You. [Электронный ресурс]. //GWT Tutorials, — Электронные данные. — Режим доступа свободный, http://code.google.com/webtoolkit/articles/dom_events_memory_leaks_and_you.html, — Заглавие с экрана. — Яз. английский.
28. Dan Morrill. Security for GWT Applications. [Электронный ресурс]. //GWT Tutorials, — Электронные данные. — Режим доступа свободный, http://code.google.com/webtoolkit/articles/security_for_gwt_applications.html, — Заглавие с экрана. — Яз. английский.
29. Dan Morrill. Using GWT for JSON Mashups. [Электронный ресурс]. //GWT Tutorials, — Электронные данные. — Режим доступа свободный, http://code.google.com/webtoolkit/articles/using_gwt_for_json_mashups.html, — Заглавие с экрана. — Яз. английский.

30. Chris Ramsdale. Large scale application development and MVP. [Электронный ресурс]. //GWT Tutorials, — Электронные данные. — Режим доступа свободный, <http://code.google.com/webtoolkit/articles/mvp-architecture.html>, — Заглавие с экрана. — Яз. английский.
31. Ларичев О. И., Петровский А. В. Системы поддержки принятия решений. Современное состояние и перспективы их развития. // Итоги науки и техники. Сер. Техническая кибернетика. — Т.21. М.: ВИНТИ, 1987, с. 131—164,
32. Сараев А. Д., Щербина О. А. Системный анализ и современные информационные технологии //Труды Крымской Академии наук. — Симферополь: СОНАТ, 2006. — С. 47-59,
33. Alter S. L. Decision support systems : current practice and continuing challenges. Reading, Mass.: Addison-Wesley Pub., 1980.
34. Bonczek R.H., Holsapple C., Whinston A.B. Foundations of Decision Support Systems.- New York: Academic Press, , 1981.
35. Thierauf R.J. Decision Support Systems for Effective Planing and Control. -Englewood Cliffs, N.J: Prentice Hall, Inc, 1982. — 536 p.
36. Sprague R. H., Carlson E. D. Building Effective Decision Support Systems. — Englewood Cliffs, NJ: Prentice-Hall, 1982.

37. Эрик Спирли. Корпоративные хранилища данных. Планирование, разработка и реализация. Издательство: Вильямс, 2001 — 400 с.
38. Rahul Biswas, Ed Ort. The Java Persistence API - A Simpler Programming Model for Entity Persistence. [Электронный ресурс]. // Oracle Technology Network. Articles, — Электронные данные. — Режим доступа свободный, <http://www.oracle.com/technetwork/articles/javaee/jpa-137156.html>, — Заглавие с экрана. — Яз. английский.
39. Daniel Rubio. Optimizing JPA Performance: An EclipseLink, Hibernate, and OpenJPA Comparison. [Электронный ресурс]. //Javalobby, — Электронные данные. — Режим доступа свободный, <http://java.dzone.com/articles/jpa-performance-optimization>, — Заглавие с экрана. — Яз. английский.
40. Jesse Smith. Learn How to Use JPA in Your Java Applications by Using Entity Manager. [Электронный ресурс]. //Javalobby, — Электронные данные. — Режим доступа свободный, <http://www.informit.com/articles/article.aspx?p=1684317>, — Заглавие с экрана. — Яз. английский.
41. Model-view-controller. [Электронный ресурс]. // Официальный сайт Wikipedia. — Электронные данные. — Режим доступа: свободный, <http://en.wikipedia.org/wiki/Model\T2A\textendashview\T2A\textendashcontroller>. —

- Заглавие с экрана. — Яз. английский.
42. Model-view-presenter. [Электронный ресурс]. // Официальный сайт Wikipedia. — Электронные данные. — Режим доступа: свободный, <http://en.wikipedia.org/wiki/Model-view-presenter>. — Заглавие с экрана. — Яз. русский.
 43. Common layers in an information system logical architecture. [Электронный ресурс] // Официальный сайт Wikipedia. — Электронные данные. — Режим доступа: свободный, http://en.wikipedia.org/wiki/Common_layers_in_an_information_system_logical_architecture. — Заглавие с экрана. — Яз. русский.
 44. Business logic layer. [Электронный ресурс] // Официальный сайт Wikipedia. — Электронные данные. — Режим доступа: свободный, http://en.wikipedia.org/wiki/Domain_layer. — Заглавие с экрана. — Яз. русский.
 45. Alex Berson, Stephen Smith, and Kurt Thearling An Overview of Data Mining Techniques. [Электронный ресурс] // Excerpted from the book Building Data Mining Applications for CRM. — Электронные данные. — Режим доступа: свободный, <http://www.thearling.com/text/dmtechniques/dmtechniques.htm>. — Заглавие с экрана. — Яз. английский.

Приложение 1. Материалы на электронном носителе