Московский Авиационный Институт
(национальный исследовательский университет)

На правах рукописи

Судаков Владимир Анатольевич

МЕТОДОЛОГИЯ УНИФИЦИРОВАННОЙ РАЗРАБОТКИ СИСТЕМ ПОДДЕРЖКИ ПРИНЯТИЯ РЕШЕНИЙ ДЛЯ МНОГОКРИТЕРИАЛЬНЫХ ВЫСОКОРАЗМЕРНЫХ ЗАДАЧ РАКЕТНО-КОСМИЧЕСКОЙ ОТРАСЛИ

Специальность 05.13.01
Системный анализ, управление и обработка информации
(информатика, управление и вычислительная техника)

Диссертация
на соискание учёной степени доктора технических наук

Научный консультант: доктор технических наук, профессор
Хахулин Геннадий Федорович

Москва — 2014 год

Содержание

Введение ... 5
Глава 1. Методы и системы многокритериального анализа альтернатив ... 20
1.1. Подходы к принятию решений ... 20
1.2. Методы многокритериального анализа альтернатив ... 22
1.3. Сравнение программных средств поддержки решений ... 27
1.3.1. Системы бизнес-аналитики ... 29
1.3.2. СППР оболочки ... 35
1.4. СППР DSS/UTES ... 38
1.4.1. Основные функции DSS/UTES ... 39
1.4.2. Средства выявления и формализации предпочтений пользователя ... 42
1.4.3. Архитектура DSS/UTES ... 44
1.4.4. Решение задачи оценивания районов размещения ракет в DSS/UTES ... 47
1.5. Выводы по первой главе ... 49
Глава 2. Практика разработки систем поддержки решений в задачах с высокой размерностью векторного критерия ... 51
2.1. «Автоматизированная система мониторинга муниципальных образований ... 51
(АСМ МО)» ... 51
2.1.1. Вычисление обобщенных показателей ... 53
2.1.2. Работа пользователя с интерфейсом АСМ МО ... 63
2.1.3. Управление на основе мониторинга ... 66
2.1.4. Программная реализация и внедрение АСМ МО ... 69
2.2. «Автоматизированная система контроля и управления (АСКУ)» ... 70
2.2.1. Правила построения обобщенных оценок на основе формальных правил ... 72
2.2.2. Программное обеспечение АСКУ ... 76
2.2.3. Работа пользователя с интерфейсом АСКУ ... 77
2.2.4. Оценка состояния войсковых частей и отделов ... 79
2.3. СППР ГЛОНАСС ... 81
2.3.1. Состав системы ГЛОНАСС ... 81
2.3.2. Сравнение DSS/UTES и ExpertChoice в задаче выбора навигационных приемников 84
2.3.3. Программное обеспечение СППР ГЛОНАСС ... 91
2.4. Выводы по второй главе ... 92
Глава 3. Разработка нового метода решения задач с высокой размерностью векторного критерия ... 93

3.1. Постановка задачи ... 93
3.2. Формализация задачи и алгоритмы решения ... 96
3.3. Дискретизация шкал критериев ... 106
3.4. Программная реализация метода ... 111
3.5. Проверка работоспособности гибридного метода построения функции предпочтений . 113
3.6. Выводы по третьей главе ... 118
Глава 4. Создание методологии унифицированной разработки СППР ... 119
4.1. Принципы унифицированной разработки ... 119
4.2. Функциональные требования к программному обеспечению СППР ... 121
4.3. Унифицированная архитектура СППР ... 125
4.4. Унифицированная модель информационного обеспечения СППР ... 128
4.5. Обеспечения качества программного обеспечения СППР ... 130
4.6. Формализация процесса разработки СППР ... 133
4.7. Разработка механизма сбора информации о процессах жизненного цикла СППР ... 140
4.8. Критерий эффективности разработки программного обеспечения СППР ... 146
4.9. Разработка OLAP системы для мониторинга процесса разработки СППР ... 150
4.10. Моделирование процесса разработки СППР ... 155
4.10.1. Цели моделирования ... 155
4.10.2. Конструирование имитационной модели ... 156
4.10.3. Выбор класса имитационной модели ... 166
4.10.4. Критерии оценки результатов моделирования ... 166
4.10.5. Модельные события ... 167
4.10.6. Атрибуты модельных событий ... 168
4.10.7. Определение системы параметров ... 169
4.10.8. Переменные состояния ... 170
4.10.9. Моделирование случайных событий, распределений и потоков ... 172
4.10.10. Сбор статистики и расчет критериев ... 174
4.10.11. Проектирование базы данных для хранения статистики ... 175
4.10.12. Разработка программного обеспечения ... 178
4.10.13. Алгоритмы обработки модельных событий ... 181
4.10.14. Имитационные эксперименты ... 191
4.10.15. Верификация модели ... 199
4.11. Выводы по четвертой главе ... 206
Глава 5. Создание СППР «Космос» на основе методологии унифицированной разработки СППР ... 207

5.1. Постановка и решение задачи планирования 209
5.2. Реализация методологии унифицированной разработки в СППР «Космос» 215
5.2.1 Регистрация в системе 215
5.2.2. Редактирование альтернатив 216
5.2.3. Редактирование критериев 220
5.2.4. Редактирование ресурсов 225
5.2.5. Выбор методов 226
5.2.6. Редактирование системы ценностей 227
5.2.7. Запуск процедуры ранжирования и мониторинг результатов 228
5.2.8. Редактирование списка планов 229
5.2.9. Запуск процедуры планирования и мониторинг результатов 231
5.3. Оценка эффекта от применения методологии 232
5.4. Выводы по пятой главе 233
Заключение 235
Список литературы 236
Приложения 250
Приложение 1. Трассировка гибридного метода 250
Приложение 2. Трассировка процедуры дискретизации шкалы 282
Приложение 3. Внедрение результатов диссертационной работы 289

Введение

Актуальность работы

В настоящее время в наиболее наукоемкой отрасли производства - *ракетно-космической отрасли (РКО)* существует ряд актуальных задач, требующих применения *систем поддержки принятия решений (СППР)* [88, 100-107, 112, 125-130, 151, 154, 158, 162]. Более того, новые задачи такого рода возникают постоянно. Задачи, решаемые в РКО с помощью СППР, характеризуются *рядом особенностей*:

- *во-первых,* разнородностью объектов для которых ставятся эти задачи и разноплановостью возникающих при этом алгоритмов их решения. Это и процесс проектирования с выбором лучших вариантов разрабатываемой системы; и процесс планирования программ выполнения комплекса работ с выбором наиболее перспективных из них; и очередность их выполнения; и применение сложных организационно-технических систем с задачами оценки эффективности функционирования составных частей и принятием решений по повышению этой эффективности; и задачи оперативного управления сложными техническими объектами с выбором рациональных вариантов решений ;
- *во-вторых*, разнородностью математического аппарата, применяемого при решении поставленных задач. Это математические модели: оптимального планирования, управления сложными динамическими объектами, многокритериальной оценки эффективности технических и организационных решений;
- *в-третьих,* особенностью коллектива людей во главе с лицом, принимающим решения (ЛПР), обладающим определенным опытом, учет которого необходим в создаваемой СППР.

Перечисленные выше особенности приводят, в конечном итоге, к необходимости создания *специализированной СППР*, ориентированной на решение данной задачи.

Несмотря на разнообразие рассматриваемых в работе задач ракетно-космической отрасли, в целом они имеют следующие *общие черты*:

- наличие ряда альтернативных вариантов возможных решений, из которых необходимо произвести выбор;
- векторный характер критерия оценки эффективности решений с высокой размерностью этого вектора (в некоторых задачах – десятки, а иногда и сотни компонентов);
- разнородность шкал отдельных компонентов векторного критерия, часть из которых

имеет количественный, а часть - лексический (качественный) характер;

- реализация решений является, как правило, ресурсоемкой (в том числе, то такому важному ресурсу, как время) и принятие неэффективных решений связано с большими потерями;
- создание самой специализированной СППР в свою очередь требует значительных ресурсов и привлечения высококвалифицированных разработчиков.

Существует развитая теория принятия решений с многочисленными методами и методиками, обладающими спектром достоинств и недостатков в определенных условиях применения. Есть достаточно много работ, связанных с методами многокритериальной оценки альтернатив. Это работы О.И. Ларичева, А.Б. Петровского, В.Д. Ногина, В.В. Подиновского, А.В. Лотова, Б.Г. Литвака, Р.Б. Статникова, В.В. Бомаса, В.Н. Козлова, Т. Саати, Р. Кини, Х. Райфа и других авторов в России и за рубежом [21-33, 67, 69, 76-85, 113-118]. Компьютерная поддержка решений на базе субъективного подхода активно развивается в ИПУ РАН Э.А. Трахтенгерцем и его учениками [151-152].

Попытки создания полностью универсальной СППР оказались неэффективными. Разработанные и реализованные СППР, во-первых, не опираются на весь спектр имеющихся методов и методик, во-вторых, не позволяют учесть опыт коллектива, осуществляющего решение задачи.

Не является продуктивным и создание специализированной СППР для каждой возникающей задачи, так сказать «с нуля». Это, как уже говорилось, с одной стороны, требует значительных затрат ресурсов и времени, привлечения квалифицированного коллектива разработчиков. С другой стороны, создание СППР должно опережать процесс решения самой задачи, для которой она разрабатывается, иначе ее использование может оказаться как минимум неэффективным, а то и вовсе бесполезным.

Решение данной проблемы может лежать в плоскости ***создания методологии*** *унифицированной разработки специализированной СППР и поддерживающих ее инструментальных средств*. Такая методология должна:

- основываться на всем спектре имеющихся методов теории принятия решений;
- позволить учесть опыт конкретного коллектива экспертов и ЛПР, ответственных за решение конкретной задачи;
- базироваться на развитых инструментальных средствах поддержки процесса создания СППР.

Таким образом, актуальной научно-технической проблемой является создание методологии унифицированной разработки систем поддержки многокритериальных решений и поддерживающих ее инструментальных средств, позволяющих объединить в создаваемых

СППР комплекс существующих методов, *эффективно (быстро и с меньшими затратами ресурсов) настраивать их на решение разнообразных конкретных высокоразмерных многокритериальных задач ракетно-космической отрасли.*

Объектом исследования являются системы поддержки многокритериальных решений, позволяющие учитывать предпочтения пользователя и работающие с векторным критерием высокой размерности.

Предмет исследования – это процесс разработки систем поддержки многокритериальных решений в высокоразмерных задачах оценки, ранжирования, выбора и оптимизации для нужд ракетно-космической отрасти.

Цели и задачи работы. Целью диссертации является создание методологии унифицированной разработки систем поддержки принятия многокритериальных решений в высокоразмерных задачах ракетно-космической отрасли и инструментальных средств поддержки этой методологии.

Для достижения выбранной цели необходимо решить следующие *задачи* -

1. Проанализировать и обобщить имеющийся опыт по созданию универсальных, а также специализированных СППР для нужд ракетно-космической отрасли, в том числе:

1.1. Выполнить мониторинг иерархических структур при целевом развитии глобальной навигационной системы, в частности, многокритериальную оценку и ранжирование навигационных приемников. (В рамках данной задачи, в ходе выполнения диссертационной работы создана СППР ГЛОНАСС).

1.2. Провести контроль сил и средств космических войск с учетом их иерархической структуры и сформировать предложения по управляющим воздействиям, улучшающим значения целевых показателей войсковых подразделений. (В ходе выполнения диссертационной работы создана «Автоматизированная система контроля и управления (АСКУ)». Для реализации данной задачи применялась так же СППР DSS/UTES).

1.3. Провести мониторинг муниципальных образований (В процессе подготовки диссертационной работы создана «Автоматизированная система мониторинга муниципальных образований (АСМ МО)», а так же дорабатывалась СППР DSS/UTES).

2. На основе опыта и анализа решения задач, указанных выше, необходимо было:

2.1. Разработать новый гибридный метод построения функции предпочтений (ФП), позволяющий выявлять систему ценностей ЛПР с учетом зависимостей по предпочтениям между компонентами векторного критерия и при высокой

размерности векторного критерия. В том числе, и метод минимизации информационных потерь при дискретизации шкал критериев.

2.2. Создать методологию унифицированной разработки специализированных СППР и инструментальные средства ее поддержки, в том числе:

2.2.1. Создать принципы проектирования СППР, позволяющие работать с векторным критерием высокой размерности, объединяющие различные методы теории принятия решений.

2.2.2. Разработать унифицированную концептуальную модель реляционной базы данных СППР, инвариантную по отношению к предметной области, позволяющую проводить настройку на конкретную задачу без изменения структуры базы данных.

2.2.3. Разработать каркасный подход к трехуровневой архитектуре программного обеспечения СППР, позволяющий проводить быструю и эффективную адаптацию оболочки СППР к решаемой задаче с учетом предпочтений ЛПР, а также проводить быстрое внедрение СППР с интеграцией в информационную среду предприятий ракетно-космической отрасли.

2.2.4. Создать набор рекомендаций, обеспечивающих получение качественного программного обеспечения СППР при решении конкретных задач.

2.2.5. Разработать инструментальные средства поддержки процесса управления разработкой СППР, в том числе имитационную модель для получения вероятностно-временных оценок производительности сотрудников занятых в разработке ПО СППР.

3. Для достижения выбранной цели, наряду с перечисленным в п.п. 1-2, необходимо так же на примере СППР «Космос» показать *эффективность применения созданной методологии* унифицированной разработки СППР. Методологии, решающей актуальную многокритериальную задачу ракетно-космической отрасли по формированию программ космических экспериментов, проводимых на российском сегменте Международной Космической Станции.

Научная новизна. Интеграция множества методов теории принятия решений в единую СППР определяет следующие новые теоретические результаты:

- создана методология унифицированной разработки систем поддержки принятия многокритериальных решений в высокоразмерных задачах ракетно-космической отрасли и инструментальные средства поддержки этой методологии, в том числе: принципы проектирования СППР, унифицированная концептуальная модель реляционной базы данных СППР, каркасный подход к трехуровневой архитектуре программного обеспечения СППР, набор рекомендаций по созданию ПО СППР. Данная методология позволяет эффективно (быстро, с минимальными затратами

ресурсов и качественно) создавать прикладные СППР;

- создан новый гибридный метод формирования функций предпочтений, являющийся частью общей методологии, существенно развивающий метод функций предпочтений в DSS/UTES, и позволяющий выявлять систему ценностей ЛПР по компонентам высокоразмерного векторного критерия с учетом зависимостей по предпочтениям между этими компонентами, новизна которого заключается в двухэтапной процедуре оценки альтернатив во всех точках пространства критериев:
    1. Декомпозиция всего критериального пространства на подобласти и определение отношения доминирования между ними качественными методами теории принятия решений (Парето-оптимальность, качественный учет важностей Подиновского, UTES, ЗАПРОС).
    2. Применение количественных методов скаляризации векторного критерия внутри сформированных подобластей;
- в составе гибридного метода разработана дискретизации шкал критериев на основе минимизации информационных потерь методом динамического программирования. Эффект от использования метода заключается в:
    1. Расширении возможностей работы с векторным критерием высокой размерности.
    2. Сокращении времени формирования функции предпочтений ЛПР.
    3. Возможности учета зависимости по предпочтениям.
    4. Отсутствии необходимости оцифровки качественных показателей.
    5. Уменьшении информационных потерь при многоуровневом агрегировании критериев;
- на основе методологии унифицированной разработки СППР создана СППР «Космос», с помощью которой решена актуальная задача автоматизации планирования научно-прикладных исследований (НПИ) в пилотируемой космонавтике. В составе СППР «Космос» разработан метод формирования и оптимизации планов НПИ на базе локальной стратегии поиска.

Практическая значимость работы. Методология унифицированной разработки позволяет достичь высокой эффективности процесса создания СППР, обеспечивая при относительно небольших затратах ресурсов и времени своевременную поддержку принятия многокритериальных решений.

Созданные автором СППР, опыт разработки которых использовался для формирования методологии, имеют самостоятельное практическое значение:

1. Разработка СППР «Космос», внедренная в Федеральном государственном унитарном

предприятии «Центральный научно-исследовательский институт машиностроения» (ФГУП ЦНИИмаш), используемая для ранжирования космических экспериментов и формирования программ НПИ (см. приложение 3 - акты сдачи-приемки научно-технической продукции № 1 от 15.08.2011, №1 от 15.10.2011; технический акт № 1 от 10.05.2012).

2. Разработка СППР ГЛОНАСС, созданная по заказу ОАО «Российские космические системы», позволяющая производить мониторинг и оценку управленческих решений при комплексном целевом развитии ГЛОНАСС (см. приложение 3 - акт сдачи-приемки научно-технической продукции №1/271 от 28.10.2011).
3. Разработка «Автоматизированной системы контроля состояния сил и средств и поддержки управляющих решений (АСКУ)», созданная в интересах Главного испытательного центра испытаний и управления космическими средствами (ГИЦИУ КС). АСКУ использовалась для оценки войсковых подразделений космических войск (см. приложение 3 - технический акт №178/1 от 15.05.2003 и акт сдачи-приемки работ №178 от 15.05.2003).
4. Разработка СППР DSS/UTES, которая решала задачи выбора полевых позиционных районов размещения ракет и используется при проведении лабораторных, курсовых и дипломных работ кафедры 302 «Автоматизированные системы обработки информации и управления» ФГБОУ ВПО «Московский авиационный институт (национальный исследовательский университет)» (МАИ).
5. Разработка «Автоматизированной системы мониторинга муниципальных образований (АСМ МО)», которая эксплуатировалась в Министерстве по делам федераций, национальной и миграционной политики Российской Федерации, а также в ряде городов: Иваново, Псков, Печоры, Дзержинский. АСМ МО использовалась для мониторинга и комплексной оценки эффективности развития муниципальных образований на региональном, областном и федеральном уровнях (см. приложение 3 - заключение № 14/2-661 от 29.01.2001 и акт сдачи-приемки научно-технической продукции по договору 36/2000 от 01.03.2000).

Методы исследования. В диссертации используются современные методы теории принятия решений, в частности методы многокритериального анализа решений: взвешенная сумма, мультипликативная свертка, идеальная точка, метод ЗАПРОС академика О.И. Ларичева, Парето-оптимальность, качественный учет важностей критериев В.В. Подиновского. Эти методы позволили создать методику выявления системы ценностей ЛПР с учетом зависимостей по предпочтениям между компонентами векторного критерия. Кроме того, использован метод дискретной оптимизации на базе локальной стратегии поиска изложенный в работах Г.Ф.

Хахулина [36, 97, 129, 141, 157]. Созданный каркасный подход к разработке СППР базируется на принципах расширяемого программного обеспечения М.М. Горбунова-Посадова [52].

На защиту выносятся следующие положения:

1. Методология унифицированной разработки СППР и инструментальные средства ее поддержки включают: принципы проектирования СППР, унифицированную концептуальную модель БД, каркасный подход к 3-х уровневой архитектуре программного обеспечения СППР, набор рекомендаций по обеспечению качества программирования СППР. Методология унифицированной разработки СППР позволяет повысить скорость адаптации СППР под требования пользователя, снижает затраты на внедрение и сопровождение СППР, облегчает интеграцию СППР с другими информационными системами.
2. Гибридный метод построения функций предпочтений, позволяющий интегрировать качественные и количественные методы на едином критериальном пространстве. Метод позволяет задавать предпочтения в высокоразмерном пространстве критериев, при этом сохраняется зависимость по предпочтениям между компонентами векторного критерия и учитывается влияние небольших изменений по непрерывным показателям на итоговую предпочтительность альтернатив. Метод позволяет определить предпочтения во всех точках критериального пространства и в дальнейшем проводить оценку произвольного количества альтернатив в автоматическом режиме. Частью гибридного метода является процедура дискретизации шкал критериев, минимизирующая информационные потери. Для этого проведена формализация задачи минимизации информационных потерь обобщенных оценок альтернатив, разработан алгоритм решения оптимизационной задачи методом динамического программирования.
3. Результаты применения созданной методологии унифицированной разработки СППР в задаче формирования программ космических экспериментов (КЭ) проводимых на российском сегменте Международной Космической Станции. На основе методологии была создана СППР «Космос». Она размещена в сети интернет по адресу http://54.214.246.161/ . Открытые исходные тексты СППР «Космос» опубликованы на сайте https://code.google.com/p/microgravity/. С помощью гибридного метода была проведена оценка критерия «Значимость КЭ», информационные потери для критерия «Прикладной эффект КЭ» были снижены на 11%. Разработка СППР «Космос» заняла около года и методология унифицированной разработки сократила сроки примерно в 3 раза.

Степень достоверности и апробация результатов. Результаты диссертации

докладывались на научных семинарах: «Фундаментальные и прикладные космические исследования» ИКИ РАН (Москва, 2012); 5th European Conference for Aeronautics and Space Sciences (Мюнхен, 2013); третья конференция «Информационные технологии на службе оборонно-промышленного комплекса России» (Саров, 2014); семинар «Казначейская система исполнения бюджета и её модели» (Москва, 2002); заседание Координационного научно-технического совета Федерального космического агентства по программам научно-прикладных исследований и экспериментов на пилотируемых космических комплексах (Королев, 2012); семинар по искусственному интеллекту кафедры «Автоматизированные системы обработки информации и управления» МАИ (Москва, 2000 г.). Материалы диссертации представлялись на следующих конференциях: «Современные технологии в задачах управления, автоматики и обработки информации» (Алушта, 1999, 2002, 2008, 2011); конференция студентов и аспирантов «Техническая кибернетика, радиоэлектроника и системы управления» (Таганрог, 2000); межвузовская научно-техническая конференции студентов и аспирантов «Микроэлектроника и информатика» (Москва, 2001); молодёжная научно-практическая конференция «Инновации в авиации и космонавтике» (Москва, 2012); Международная конференция по неравновесным процессам в соплах и струях (Алушта, 2012); Международная конференция по Вычислительной механике и современным прикладным программным системам (Алушта, 2013), Международная конференции «Авиация и космонавтика» (Москва, 2013).

Работа поддержана Грантом Президента Российской Федерации № МК-5232.2007.9, грантом РФФИ № 3-01-00895 А, Госконтрактом № (105-1313-2011)-1313/140-2012-1 (№ гос. регистрации У92182), НИР «Комплекс – МАИ», НИР “ПРОСТОР-Р”.

Публикации. Основные результаты диссертации представлены в 2 монографиях, 10 научных статьях, опубликованных в журналах, входящих в перечень ВАК. В РОСПАТЕНТе зарегистрированы 2 программы для ЭВМ. Помимо этого, результаты опубликованы в других журналах, сборниках статей и трудах конференций на русском и английском языках. Общее число публикаций — 34.

Личный вклад. Все основные результаты, изложенные в диссертации, включая постановки задач и их алгоритмические решения и созданное программное обеспечение, получены автором лично или выполнены под его научным руководством и при непосредственном участии.

В первой главе рассмотрены основные методы многокритериального анализа альтернатив и СППР, созданные по принципу оболочек. Исследуются их достоинства и недостатки. В том числе, исследованы методы:

- взвешенная сумма;

- мультипликативная свертка;
- свертка Гермейера;
- метод идеальной точки;
- медиана Кемени;
- метод последовательных уступок;
- оптимальность по Парето;
- качественный учет важностей критериев В.В.Подиновского;
- Метод Анализа Иерархий;
- ЗАПРОС;
- функции полезности;
- функции предпочтений в DSS/UTES;
- ПРИНН.

Проанализированы простота понимания, возможности ввода и редактирования системы ценностей ЛПР и трудоемкость этих процедур, выявлены границы применимости метода.

Среди СППР, созданных по принципу оболочек, проанализированы следующие системы, которые представлены на коммерческом рынке и успешно эксплуатируются: Expert Choice, Super Decisions, Criterium Decision Plus, Decision Lens, ИАС “ОЦЕНКА и ВЫБОР”, «Быстрый прототип», DASS, FEASIBLE GOALS, ПК MOVI, Web-HIPRE. Выделены СППР, основанные на оценке полезности, программные комплексы, разработанные в Рижском политехническом институте под руководством проф. А.Н. Борисова: DELTA, SDMIS, "Альтернатива". Рассмотрена СППР DSS-UTES, предназначенная для поддержки многокритериальных решений в сложной информационной среде. Рассмотрена СППР DSS/UTES, которая апробировалась в задаче многокритериального оценивания районов размещения ракет.

В второй главе рассмотрен практический опыт создания конкретных СППР, применявшихся в высокоразмерных задачах ракетно-космической отрасли. Анализ требований к конкретным СППР, систематизация проектных решений по архитектуре СППР, обобщение практического опыта алгоритмизации и программирования различных СППР, созданных при непосредственном участии автора в качестве главного проектировщика и разработчика, позволил в дальнейшем сформулировать методологию унифицированной разработки СППР для решения высокоразмерных задач ракетно-космической отрасли.

Возможности СППР DSS/UTES были сопоставлены с возможностями американской СППР ExpertChoice, созданной при участии видного ученого Т. Саати, в задаче оценки навигационных приемников ГЛОНАСС.

На базе СППР DSS/UTES создана «Автоматизированная система мониторинга муниципальных образований (АСМ МО)», которая позволяет осуществлять мониторинг

муниципальных образований по большой иерархии показателей. Общее количество исходных показателей превышало 200. АСМ МО создавалась по заказу Министерства по делам федераций, национальной и миграционной политики Российской Федерации в рамках федеральной программы «Совершенствование управления сферой социально-экономического развития муниципальных образований на основе разработки и внедрения современных информационных технологий».

В интересах Главного испытательного центра испытаний и управления космическими средствами (ГИЦИУ КС) была разработана «Автоматизированная система контроля и управления (АСКУ)». АСКУ предназначена для мониторинга состояния иерархических систем по многим показателям, имеющим древовидную структуру. Древовидный характер структуры показателей, с одной стороны, определяется иерархической структурой системы, а, с другой стороны, древовидным характером показателей для каждого объекта системы. Объекты на одном уровне иерархии могут быть разнотипными, т.е. дерево показателей для них может быть своё. Дерево показателей для каждого объекта иерархической структуры определяется пользователем АСКУ. Отдельные показатели могут быть как числовыми, так и лексическими. Корень всей древовидной структуры показателей соответствует обобщенной оценке состояния всей оцениваемой системы, а листва представляет так называемые исходные показатели. Они не являются обобщенными, вычисляемыми, а задаются пользователем как исходные данные для оценки состояния всей системы и отдельных её частей (объектов). Общее количество исходных показателей для оценки структурных подразделений космических войск было порядка 100. Таким образом, задачи, решаемые АСКУ, были высокоразмерными.

Сложность структуры системы ГЛОНАСС, многообразие возможных направлений ее развития, многокритериальный характер оценки эффективности ее компонент и системы в целом, ограниченность выделяемых на развитие и эксплуатацию системы ресурсов определяет актуальность разработки СППР, позволяющей осуществить сравнение альтернативных вариантов развития и выбор наилучших.

Возможностей СППР DSS/UTES по хранению и обработки больших информационных массивов для работы с системой ГЛОНАСС в комплексе уже недостаточно. Поэтому была поставлена задача разработки новой СППР ГЛОНАСС на базе современных информационных технологий. СППР ГЛОНАСС предназначена для поддержки решений на всех этапах развития и целевого использования глобальной навигационной системы ГЛОНАСС. Возможности СППР ГЛОНАСС были также продемонстрированы на примере задачи оценки навигационных приемников. Программное обеспечение СППР ГЛОНАСС создавалось с использованием свободного системного программного обеспечения (ПО).

Третья глава посвящена разработке новых методов решения многокритериальных задач с высокой размерностью векторного критерия. Решаемые в работе практические задачи из ракетно-космической отрасли характеризуются десятками критериев, имеют зависимости по предпочтениям между критериями, содержат лексические критерии, оцифровка которых связана со значительными трудностями.

В первой части третьей главы предложен гибридный метод построения функций предпочтений. Метод создан с целью решения следующих задач:

- обеспечить ЛПР инструментарием для качественных суждений о предпочтительности альтернатив в задачах с высокой размерностью векторного критерия. (Под качественными понимаются суждения в порядковой шкале - ЛПР может сказать, что одно решение лучше другого; ЛПР может указать, что данный критерий важнее другого; но при этом получить числовые оценки степени превосходства затруднительно);
- выявить предпочтения ЛПР во всем критериальном пространстве, чтобы дальнейшая оценка конкретных альтернатив проходила в автоматическом режиме. (Таким образом, процесс выбора и ранжирования альтернатив происходит быстро и появляется возможность проводить процедуру поиска решений без привлечения ЛПР, но с учетом его предпочтений);
- учесть зависимости между компонентами векторного критерия по предпочтениям. (В результате можно устранить ситуации, когда альтернативы с неприемлемыми для ЛПР оценками по одному критерию, получают высокую интегральную оценку за счет других критериев);
- обеспечить различимость альтернатив в случае когда значения критериев подвергаются искусственной дискретизации с целью замены непрерывных шкал на бальные оценки.

Метод носит названия гибридного, так как сочетает элементы качественных методов (функции предпочтений, качественные важности критериев, метод ЗАПРОС) и количественных методов (взвешенная сумма, идеальная точка) выявления суждений ЛПР.

Итоговая оценка альтернативы, полученная указанным методом, состоит из целой части, отражающей качественные суждения пользователя, и дробной части, отражающей применение количественных методов, с целью повышения чувствительности и возможности оптимизации на непрерывном множестве. (Результат построения ГФП в задаче оценки космических экспериментов проводимых на борту МКС показан в пятой главе работы).

Во второй части третьей главы рассматривается задача построения многоуровневого дерева критериев. В этом подходе показатели объединяются в группы, и каждая из них

характеризуется своим показателем называемым *обобщенным.* Формирование групп должно происходить с участием или под контролем ЛПР. Для каждого обобщенного показателя задается правило его вычисления (решающее правило). В качестве решающего правила может выступать формальный метод свертки или процедура свертки по предпочтениям, в результате которой вычисляется «промежуточная ФП».

Дерево агрегирования во многом напоминает аналитическую иерархию из метода анализа иерархий. Отличие заключается в том, что для свертки показателей, которые могут быть как числовыми, так и лексическими, в первую очередь, используется свертка по предпочтениям.

При агрегировании показателей мы получаем высокую размерность шкал обобщенных показателей. Назначение предпочтений на таких шкалах затруднительно, поэтому в работе предложена процедура дискретизации шкал обобщенных показателей. Она содержит две подзадачи. Первая - выбор числа градаций, и вторая - разбиение шкалы на градации. Шкалу полученную в результате такой процедуры назовем «загрубленной». При решении первой подзадачи пользователь, на основе содержательных соображений, сам назначает требуемую точность оценки. Для решении второй подзадачи предлагаются три способа:

1. Разбиение на равные интервалы. В информационном плане, как будет показано далее, этот метод не всегда целесообразен.
2. Предоставление пользователю возможности самому формировать интервалы разбиения на основе содержательного анализа. Неудобство заключается в том, что пользователю системы придется тратить достаточно много времени на разбиение и согласование шкал.
3. Разбиение шкалы обобщенного показателя на основе минимизации информационных потерь.

Очевидно, что дискретизация шкал приводит к потерям информации. Наиболее полная информация о векторном критерии содержится на уровне исходных показателей. Потери возрастают от листвы к корню дерева агрегирования. Оценить потери информации, возникающие в процессе агрегирования показателей, можно с помощью энтропии. Как известно, количество информации измеряется разностью энтропий до и после получения сообщения. В нашем случае энтропия показателя после получения сообщения равна нулю и количество информации в сообщении определяется энтропией показателя до его получения. Поэтому осуществлять разбиение шкалы обобщенного показателя на градации желательно так, чтобы энтропия показателя была максимальной. А это приводит к идее разбиения на такие интервалы, в которых сообщения появлялись бы с одинаковыми вероятностями. Для

максимизации энтропии показателя поставлена и решена оптимизационная задача, которая относится к классу задач дискретного динамического программирования.

В работе проведен расчет выигрыша от оптимального разбиения в задаче оценки прикладного эффекта космического эксперимента (КЭ) планируемого для поведения на российском сегменте МКС. Прикладной эффект КЭ – востребованность (хотя бы потенциальная) эксперимента министерствами, ведомствами, частными организациями и т.п. Прикладной эффект определяется на основе 8 показателей, каждый из которых оценивается по 3-х бальной шкале. Шкала предпочтений прикладного эффекта от КЭ изначально составляла 256 градаций. Необходимо было получить «загрубленную» шкалу из 16 градаций. Для определения p_k использовались данные по оценкам прикладного эффекта по 93 космическим экспериментам. Информационные потери при равномерном разбиении шкалы загрубляемого критерия составляют 28 %, в то время как потери в оптимизированной шкале составили 17 %.

В четвертой главе рассматриваются принципы унифицированного подхода к разработке СППР. По мнению автора, основная потребность в унификации связана со стремлением сделать максимально широким и доступным весь набор информационных и аналитических ресурсов и сервисов для ЛПР. Современные методы системного и информационного моделирования бизнес процессов (например технология SADT, язык UML), методы проектирования и системной интеграции информационно-аналитических систем дают возможность не только формализовать с использованием унифицированных подходов основные процедуры выработки и обоснования решений, рекомендуемые теорией, но и предложить решение по программной реализации ресурсов и сервисов СППР, инвариантных по отношению к предметной области.

Унифицированный подход к разработке программного обеспечения СППР состоит из:

- совокупности принципов проектирования;
- каркасного подхода к разработке архитектуры программного обеспечения;
- шаблона правил качественного кодирования;
- набора характеристик оценки качества программного обеспечения СППР.

Унификация заключается в том, что все эти методические рекомендации применимы вне зависимости от того, в какой предметной области необходимо решать задачи поддержки решений. Конечно, некоторые предметные области могут потребовать специфичных проектных решений, но они не противоречат предложенному унифицированному подходу, а лишь дополняют его.

В пятой главе рассмотрены вопросы информатизации и автоматизации формирования Этапной программы (ЭП) научно-прикладных исследований и экспериментов, планируемых на российском сегменте МКС. Осуществлена разработка СППР «Космос» для определения

приоритетности космических экспериментов и разработана методика составления плана на основе предпочтений пользователя. СППР «Космос» использует все методологические рекомендации по разработке программного обеспечения СППР изложенные в четвертой главе, а также унифицированную концептуальная модель реляционной БД СППР.

В настоящее время, в связи с увеличением количества предложений по новым космическим экспериментам (КЭ), а также КЭ, находящимся в стадии подготовки и реализации, актуальным является повышение эффективности экспертных процедур при выявлении приоритетов и формировании ЭП научно-прикладных исследований и экспериментов, планируемых на российском сегменте (РС) МКС. Достигнуть этого возможно путем применения информационных, в том числе сетевых, технологий на стадии принятия решений о включении проектов в ЭП.

На базе СППР DSS/UTES была создана новая СППР «Космос» позволяющая определять приоритетность КЭ. Работа с ней состоит из следующих этапов:

- ввод векторного критерия для оценки КЭ;
- ввод экспертами оценок конкретных КЭ по заданному векторному критерию;
- ввод системы ценностей, отражающей важности конкретных компонент векторного критерия (с помощью метода парных сравнений, зависимых предпочтений или иными формальными методами);
- вычисление обобщенных оценок КЭ на базе введенной информации в автоматическом режиме.

Для вычисления обобщённых оценок КЭ применяются как формальные методы на основе взвешенной суммы и парных сравнений, так и гибридный метод задания предпочтений.

Обобщенные оценки альтернатив позволяют перейти к следующему этапу работы с системой – это составление плана КЭ. Исходными данными для задачи служат ресурсные ограничения и ранжированный по значению обобщенной оценки список космических экспериментов, полученный в результате обработки экспертных оценок.

На заданном временном периоде необходимо спланировать последовательность проведения множества экспериментов на МКС, выбрав их из заданного множества возможных экспериментов и определив моменты их начала.

Составленный план должен удовлетворять всем ресурсным ограничениям и быть оптимизирован по векторному критерию, включающему следующие составляющие:

1. Максимум суммарной обобщенной оценки включенных в план экспериментов.
2. Минимум средневзвешенного времени начала экспериментов.

Данная задача планирования относятся к классу многокритериальных нелинейных задач дискретного целочисленного программирования. Наиболее подходящим методом решения

данной задачи является метод неявного перебора, основанный на локальной стратегии поиска. Этот метод, хотя и не гарантирует нахождения точного оптимального решения, как показывает опыт его использования, но позволяет находить с приемлемыми вычислительными затратами субоптимальные решения. Достоинством метода является возможность эффективного использования эвристик, учитывающих свойства решаемой задачи, при нахождении допустимых решений, при определении области перебора в районе рассматриваемой точки. Кроме того, существует возможность использования случайного механизма для реализации необходимой степени разнообразия при формировании начальных допустимых решений, что повышает вероятность выхода на оптимальное либо близкое к нему решение.

СППР «Космос» разработана в соответствии с методологией унифицированной разработки систем поддержки принятия многокритериальных решений. С СППР «Космос» должен работать широкий круг экспертов, который рассредоточен территориально по множеству городов и организаций. Поэтому СППР должна допускать сетевую модель работы в Intranet или Internet. Необходимо так же минимизировать затраты на установку программного обеспечения на компьютеры пользователей. Для этого целесообразно использовать кроссплатфоменного тонкого клиента, т.е. клиента допускающего работу на компьютере с различными операционными системами. Кроме того, важно обеспечить соответствие HTML кода, передаваемого на клиента, последнему стандарту HTML5. Данным требованиям в полной мере удовлетворяет веб-браузер Mozilla Firefox, поэтому он и был выбран в качестве тонкого клиента. Веб-сервер работает под управлением IIS. Серверное приложение написано в среде Microsoft VisualStudio 2005 на C# с использованием ASP.NET. Программное обеспечение реализует предложенную унифицированную архитектуру, написано с использованием каркасного подхода и набора рекомендаций по обеспечению качества СППР. Взаимодействие с базой данных организовано исключительно через хранимые процедуры. Таким образом достигается оперативность изменений бизнес-логики, повышается производительность и гибкость, обеспечивается более высокий уровень безопасности. Физическая структура базы данных соответствует унифицированной концептуальной модели, предложенной в четвертой главе. В качестве СУБД использован MS SQL Server.

В заключении изложены основные научные и практические результаты, полученные в диссертационной работе.

Глава 1. Методы и системы многокритериального анализа альтернатив

1.1. Подходы к принятию решений

Существует два разных подхода к принятию решений. Первый подход основан на создании системы правил, на основании которых система предлагает некий вариант решения. Такой подход реализуется в экспертных системах [108-109]. Если проблема хорошо структурирована, система правил достаточно четко задана и объективно является правильной, то целесообразно применение экспертных систем. Однако высокоразмерные задачи выбора и оптимизации в ракетно-космической отрасли относятся к классу слабоструктурированных. То есть на сегодняшний день нет объективного и исчерпывающего списка правил, по которым оценивается космический эксперимент, выбирается полевой позиционный район размещения ракет, проводится мониторинг космических сил и средств. Кроме того, оценки объектов ракетно-космической отрасли носят как числовой, так и качественный (лексический) характер. Для решения слабоструктурированных задач применяются системы относящиеся к другому классу – это системы поддержки принятия решений (СППР). Заметим, что общим свойством большинства рассматриваемых ниже методов заложенных в СППР является их зависимость от субъективизма лица, принимающего решения (ЛПР). Считается, что ЛПР обладает опытом, знанием и навыками, делающими его компетентным в решаемой задаче. Это проявляется в том, что выбор метода и назначение его необходимых внутренних параметров осуществляется (или, по крайней мере, должно осуществляться) либо непосредственно ЛПР, либо с его участием. Это положение приобретает принципиальный характер, когда речь идет о принятии решений с привлечением СППР [100].

Есть достаточно много работ, связанных с методами многокритериальной оценки альтернатив. Это работы О.И. Ларичева, А.Б. Петровского, В.Д. Ногина, В.В. Подиновского, А.В. Лотова, Б.Г. Литвака, Т. Саати, Р. Кини, Х. Райфа и других авторов в России и за рубежом [6, 9, 11-17, 20-33, 47, 113-117]. Компьютерная поддержка решений на базе субъективного подхода активно развивается в ИПУ РАН Э.А. Трахтенгерцем и его учениками [104]. Однако большинство СППР используют только один метод многокритериального анализа решений и не учитывают другие подходы. В некоторых СППР их разработчики закладывают свой субъективизм, который ЛПР может и не осознавать, или он может не стыковаться с предпочтениями ЛПР. На практике востребованы комплексные решения, не хватает интегрированных подходов и формализованных процедур, которые позволят обоснованно выбирать тот или иной многокритериальный метод в процессе работы системы, исходя из предпочтений ЛПР.

Следует отметить, что характерной чертой многих исследований является желание их авторов создать математическую модель человека, работающего в составе человеко-машинного комплекса. К сожалению, такие модели, как правило, не адекватны, особенно, если речь идет о многокритериальных задачах и слабо структурированной предметной области, т.к. по существу они (модели) отражают попытки формализации не изученных или, в лучшем случае, плохо изученных процессов человеческого мышления. Это положение принципиально. Принятие его требует поиска иных подходов к разработке человеко-машинных систем. Сказанное представляется особенно важным, когда объектом разработки и эксплуатации являются сами СППР. Поэтому понятен большой интерес к интеллектуальным системам поддержки решений. Их спектр весьма широк: информационные, информационно-советующие, генерирующие, оценивающие, ранжирующие и оптимизирующие альтернативные варианты решений и т.п. Подавляющее большинство СППР ориентировано на многокритериальные задачи, т.е. задачи в которых решения оцениваются и принимаются на основе векторного критерия. Принципиальная трудность таких задач заключается, как известно, в противоречии между желанием как можно подробнее описать сущность задачи и отсутствием строгих математических методов сопоставления векторов. Используемые методы так называемой «оптимизации по векторному критерию» часто сводятся по существу к его скаляризации. Таких методов достаточно много, как и различных СППР, их реализующих. Однако мало известны работы по их сравнительной оценке, на основании которых можно было бы сделать выбор СППР. В данной работе, помимо прочего, делается попытка восполнить этот пробел. С этой целью рассматривается на одном и том же материале решение задачи ранжирования альтернатив с использованием наиболее известных способов скаляризации векторного критерия и осуществляется их сравнительная оценка.

СППР помогает в принятии обоснованных и эффективных решений, направленных на экономию ресурсов ракетно-космической отрасли, поэтому актуальными являются научно обоснованные решения в области разработки СППР, внедрение которых вносит значительный вклад в развитие страны.

Неоднократные попытки моделировать деятельность человека в составе человеко-машинного комплекса (включая оценку ситуации и формализацию целей и стратегий) на основе традиционных подходов показали их несостоятельность, в первую очередь из-за слабой структурированности решаемых задач в целом. Под слабой структурированностью понимается преобладание качественных (лексических) критериев и качественных суждений ЛПР. Кроме того, необходимо постоянно иметь в виду, что инструментальные СППР должны ориентироваться на поддержку решений, а отнюдь не на их окончательное принятие. Последнее является эксклюзивной функцией лица принимающего решения (ЛПР) и должно в полной мере

отражать его систему ценностей. В силу этого, интерес разработчиков СППР смещается в сторону создания статистических (например, нейронные сети), лексикографичесих моделей, и моделей, основанных на выявлении и формализации полезности и предпочтений ЛПР.

1.2. Методы многокритериального анализа альтернатив

Сопоставим некоторые из этих методов, которые в первую очередь будут реализованы в СППР, в содержательном плане, опуская их подробное описание, т.к. они многократно излагались в литературе [1, 6, 9, 11, 13, 15, 32, 33, 76-79, 85, 90, 94, 99, 134].

Критерий среднего взвешенного. В этом методе обобщенный критерий эффективности представляется в виде суммы значений показателей с весовыми коэффициентами, которые также называются коэффициентами важности. Коэффициент важности отражает ценность показателя (его важность) по сравнению с остальными.

Достоинства:

- простота формализации;
- ясный физический смысл;
- широкое освещение в литературе и наличие апробированных программных средств, реализующих метод;
- возможность учета индивидуальных представлений ЛПР о задаче при назначении весовых коэффициентов (важностей);
- наличие простой формальной процедуры (метод парных сравнений), облегчающей процесс вычисления весовых коэффициентов.

Недостатки:

- неявная взаимная компенсация показателей, которая становится неконтролируемой при большом их числе;
- не учет нелинейной зависимости весовых коэффициентов от значений показателей (не учет так называемой зависимости по предпочтениям: важности показателей вводятся один раз и в дальнейшем играют роль параметров);
- неоднозначность отображения точек критериального пространства на множестве возможных решений (альтернатив).

Метод идеальной точки. В этом методе векторный критерий рассматривается как точка в n-мерном пространстве, координаты которой соответствуют значениям показателей. Наилучшим из возможных является решение, при котором по каждому из показателей будет достигнуто наилучшее значение. Соответствующая ему точка называется «идеальной». Введя в критериальном пространстве метрику, т.е. задав способ измерения расстояния между точками, логично считать ту из двух альтернатив лучшей, которая ближе к «идеальной» точке. Следует

иметь в виду, что выбор метрики может существенно сказаться на результате решения задачи и поэтому этот этап достаточно принципиален. Обычно, во имя простоты, используют евклидово расстояние.

Достоинства:

- как и в критерии среднего взвешенного, компоненты векторного критерия рассматриваются в совокупности;
- четкая формальная постановка.

Недостатки:

- произвольный выбор метрики (как правило, используется декартово пространство);
- непредставимость расстояния между двумя точками n-мерного пространства при n>3;
- неоднозначность отображения точек критериального пространства на множестве возможных решений.

Медиана Кемени. Этот метод основан на ранжировани конечного множества альтернатив. На отношениях линейного порядка (ранжированиях) вводится мера близости ранжирований, предложенная Кемени. Ранжируем альтернативы по каждой компоненте векторного критерия и для полученных ранжирований вычисляем медиану Кемени.

Достоинства:

- компоненты векторного критерия рассматриваются в совокупности;
- четкая формальная постановка.

Недостатки:

- непредставимость среднего ранжирования в содержательном плане, т.к. оно, как правило, не отвечает ни одной из экспертных (осредняемых) ранжировок.

Метод последовательных уступок. Суть этого метода состоит в следующем. Вначале показатели ранжируются по важности. Но их упорядочение носит чисто качественный характер, т.е. никаких количественных оценок важностей не производится. Затем выбирается первый самый важный показатель и находится оптимальная по нему альтернатива. После этого назначается уступка, т.е. интервал, в котором могут варьироваться значения первого показателя. Другими словами, определяется насколько первый показатель может отличаться от своего оптимального значения (насколько его можно ухудшить). Потом производится оптимизация по второму показателю. При этом оптимальное значение второго показателя ищется при допустимой уступке первого. Далее определяется уступка по второму показателю и т.д. В качестве оптимальной принимается альтернатива, вычисленная в конце многоэтапной оптимизации.

Достоинства:

- содержательная простота;

- учет всех компонент векторного критерия.

Недостатки:

- необходимость предварительного ранжирования показателей по важности;
- трудность определения величин уступок;
- практическая не реализуемость при большом числе показателей.

Строго говоря, последовательные уступки не метод скаляризации векторного критерия, а эвристический способ или процедура выбора «подходящего» решения в многокритериальной задаче.

Оптимальность по Парето. Данный метод, рассматривая все множество альтернатив, отбрасывает те из них, которые доминируются хотя бы одной альтернативой. Считается, что Альтернатива А1 доминирует над альтернативой А2, если по всем показателям (локальным критериям) А1 не уступает А2, а хотя бы по одному из них лучше. Множество недоминируемых альтернатив называется Парето оптимальным, и именно из него следует выбирать решение. Итак, поиск решений по принципу Парето оптимальности дает множество допустимых решений, а не одно единственное.

Достоинства:

- метод математически строг;
- выделяет множество допустимых решений;
- дает возможность ЛПР сосредоточить анализ решений на более узком (точнее, менее мощном) множестве альтернатив.

Недостатки:

- применимость метода ограничена мощностью Парето-оптимального множества (количество его элементов не должно превышать 7-10). Если у недоминируемого множества большая мощность, то выбор на нем без применения специальных процедур практически невыполним.

Качественный учет важностей критериев В.В. Подиновского. Данный метод, как и оптимальность по Парето, использует понятие доминирования альтернатив. Отличие заключается в расширенном толковании понятия доминирования. Все критерии в методе Полиновского упорядочиваются по важности. Пусть даны две альтернативы А1 и А2, которые отличаются только по значениям 2-х критериев. При этом А1 может быть получена из А2, если значения этих 2-х критериев переставить. Тогда А1 доминирует над А2, если для более важного критерия у А1 значение лучше, чем у А2.

Достоинства:

- метод математически строг;
- учитывает систему ценностей ЛПР при упорядочивании критериев по важности. При

этом не требуется количественная оценка этой важности;

- выделяет множество допустимых решений, которое содержит не больше решений, а обычно содержит меньше решений, чем в методе оптимальности по Парето.

Недостатки:

- на практике число недоминируемых альтернатив велико, поэтому окончательный выбор пользователем может быть затруднен;
- требуется дискретизация шкал непрерывных критериев, причем на равное количество интервалов; метод не позволяет отличать альтернативы, которые попадают в одну градацию.

Метод анализа иерархий. Метод является расширением метода среднего взвешенного. Исходный векторный критерий агрегируется не в один скалярный показатель, а вводится несколько промежуточных показателей, которые агрегируют непересекающиеся подмножества компонент векторного критерия. Эти промежуточные показатели снова агрегируются в новые показатели и процесс продолжается до выхода на один показатель, который уже характеризует альтернативное решение в целом. В результате получается дерево показателей.

Достоинства:

- позволяет работать с векторным критерием произвольной размерности;
- в методе предложена процедура вычисления весов критериев на базе парных сравнений;
- оценки по исходным показателям могут быть получены путем парных сравнений альтернатив.

Недостатки:

- не учитывает зависимости по предпочтениям;
- не позволяет оценивать промежуточные показатели в лексических шкалах;
- при большом количестве промежуточных уровней агрегации возможны потери информации.

Функции предпочтений в DSS/UTES. Предпочтения определяются в шкале отношений. Обычно используется бинарная шкала. ЛПР сопоставляет попарно совокупности значений показателей (альтернативы) и определяет, какая из альтернатив предпочтительней или же они равноценны. Таким образом, на множестве альтернатив вводится отношение не строгого порядка, что отвечает их не строгому ранжированию. Многомерная скалярная функция, формализующая это ранжирование, называется функцией предпочтений (ФП) и на ее основе возможно проводить оптимизацию и ранжирование альтернатив. Функцию предпочтений можно определить как ранжирование полного множества альтернатив по предпочтениям

пользователя. Процедуру вычисления ФП можно рассматривать как способ скаляризации векторного критерия.

Достоинства:

- учитывает зависимость по предпочтениям;
- не требует оцифровки лексических показателей;
- прост в изучении пользователем.

Недостатки:

- размерность векторного критерия не может превышать 5;
- процедура ввода предпочтений трудоемка;
- нет возможности визуализировать всю функцию предпочтений на одном экране для размерности векторного критерия больше 2;
- метод не чувствителен к изменениям по непрерывным показателям в рамках одной градации.

Существует еще достаточно много других методов многокритериального анализа решений: мультипликативная свертка, свертка Гермейера, расстояние Чебышева, ЗАПРОС, ЭЛЕКТРА, ПРИНН, метод аналитических сетей и многие другие. Основные методы показаны на рисунке 1.1. Рассмотренные методы можно разбить на 2 группы. Первая группа – это качественные методы; они позволяют работать с лексическими критериями без оцифровки, но они не позволяют работать с высокоразмерным критерием. Вторая группа – это количественные методы; в них требуется оцифровка лексических критериев и нет возможности учесть зависимости критериев по предпочтениям. Разработчики СППР не должны настаивать на выборе какого-либо одного метода. Главное, чтобы программная реализация СППР содержала инструментарий для быстрой интеграции методов, которые необходимы ЛПР.

Многие методы в теории не ограничивают размерность критериального пространства, но на практике обычно не применимы более чем к 7±2 критериям [95]. Как показано в работах Г. Саймона - это обусловлено особенностями человеческого мозга по анализу больших информационных массивов [182]. В качественных методах, например, при Парето-оптимальности, на практике, при большом количестве критериев недоминируемых альтернатив оказывается достаточно много.

Рис. 1.1. Методы многокритериального анализа альтернатив.

1.3. Сравнение программных средств поддержки решений

Для успешного решения задач поддержки решений и всех рисков, возникающих в процессе разработки и эксплуатации системы, необходимо рассмотреть концепции современных СППР и реализации этих концепций.

Под термином *система поддержки принятия решений* (СППР) обычно понимается инструментарий выработки рекомендаций для лица, принимающего решение на основе ранжирования конечного множества альтернатив (решений) или оптимизации их на бесконечном множестве [76, 151]. Сейчас под СППР часто понимают и инструментарий подготовки данных для ЛПР. Этот инструментарий помогает создавать базы данных (как правило, объемные и содержащие сложные взаимосвязи), организовывать гибкий и удобный доступ к ним через мощные средства формирования запросов, получать результаты запросов в форме, максимально удобной для последующего анализа, использовать мощные генераторы отчетов и т.п.

СППР помогают решать задачи, которые традиционно считаются «интеллектуальными». Это диагностика проблемных ситуаций из различных областей (медицина, экология, геология и т.д.), конструирование, принятие политических, экономических и других решений. Такие

задачи существовали всегда, но сегодня, благодаря развитию компьютерных систем, их решение стало практически возможным на основе формальных методов. Основные трудности были связаны с необходимостью одновременного учета множества взаимосвязанных факторов, влияющих на решение задачи, что приводило к большим информационным массивам, которые нельзя анализировать «вручную». Кроме того, широкое практическое внедрение СППР сдерживалось тем, что не все организации, которые хотели бы использовать эти системы, обладали достаточными информационно-вычислительными ресурсами. Удешевление аппаратного обеспечения, появление мощных систем управления базами данных (СУБД) и инструментальных средств разработки программ изменило ситуацию. Сейчас заказчики подобных систем в большинстве своем обладают или уже существующей информационной базой, хранящейся в ЭВМ, или достаточными финансовыми ресурсами для её создания [1, 41, 56, 86, 91, 98].

Как известно, отличительной особенностью слабо структурированных задач, т.е. именно тех задач, которые в наибольшей мере требуют привлечения СППР, является многокритериальность, заключающаяся в том, что качества принимаемых решений нельзя оценить с помощью единственного скалярного показателя и приходится прибегать к векторному критерию. Для разрешения возникающих при этом трудностей ранжирования и оптимизации альтернатив приходится использовать неформальные методы скаляризации, опирающиеся на суждения ЛПР.

Существующие многокритериальные СППР обычно делят на специализированные и оболочки. Первые обладают тем недостатком, что создание конкретной специализированной СППР каждый раз требует больших временных и соответственно финансовых затрат. Кроме того, в них обычно жестко прописан алгоритм, либо структура решения задачи, которые, как правило, более отражают взгляд на задачу разработчика системы, чем будущего пользователя (ЛПР).

В отличие от специализированных систем оболочки инвариантны по отношению к предметной области. Приобретение инвариантной СППР может быть выгодней в эксплуатационном и финансовом плане, чем заказ на изготовление специализированной. При подобном решении нужно также учитывать стоимость интеграции СППР и управляемой системы. Исключение может составлять случай, когда СППР для данной предметной области уже создана и нет необходимости решать задачи из других областей. Но даже и в этой ситуации специализированная СППР не всегда будет лучше. Так, специализированная СППР в области экономики потребует серьезной перенастройки и адаптации своих частей в случае изменений в законодательстве.

1.3.1. Системы бизнес-аналитики

Сегодня особенно активно развито и продолжает развиваться направление BI (Business intelligence – бизнес-анализ) систем [18, 46, 72, 86, 100, 110, 142-144, 160], которые зачастую относят к СППР.

BI в широком смысле слова определяет:

- процесс превращения исходных данных в информацию и знания о бизнесе для поддержки принятия улучшенных и неформальных решений;
- информационные технологии (методы и средства) сбора данных, консолидации информации и обеспечения доступа бизнес-пользователей к знаниям;
- знания о бизнесе, добытые в результате углубленного анализа детальных данных и консолидированной информации.

Это достаточно общие слова, которые нуждаются в пояснениях.

В основе технологии BI лежит организация доступа пользователей и анализ структурированных количественных данных (информации о бизнесе). BI способствует формированию заключений, нахождению взаимосвязей в целях эффективного развития предприятия. Пользователи BI это в основном руководители и аналитики.

Технология BI определяет методы и средства доступа и оперативного анализа информации в терминах предметной области. Средства BI не обязательно должны работать в инфраструктуре собственного хранилища данных. В этом случае проблема очистки и согласования данных возлагается на них. На практике зачастую выделяют транзакционную и аналитическую составляющие с целью улучшения характеристик безопасности, управляемости и удобства использования каждой их двух компонент. Основные стыки идут как на уровне информации, так и на уровне метаданных (описание предметной области в хранилищах информации). В качестве средств сбора и хранения данных используют методологию витрин данных (Data warehousing), которая определяет подходы и обеспечивает интеграцию, очистку, ретроспективное хранение информации, предназначенной для анализа, и всесторонне отвечает на вопрос подготовки информации для анализа. В этом случае можно обеспечить централизованное управление метаданными.

Сегодня категории продуктов BI включают BI-инструменты и BI-приложения. Первые, в свою очередь, делятся на:

- генераторы запросов и отчетов;
- инструменты оперативной аналитической обработки (Online Analytical Processing, OLAP);
- корпоративные BI-наборы (Enterprise Business Intelligence Suites, EBIS);

- BI-платформы.

Большинство инструментов BI применяются конечными пользователями для доступа, анализа и генерации отчетов по данным, которые чаще всего располагаются в хранилище, витринах данных или оперативных складах данных. Разработчики приложений используют BI-платформы для создания BI-приложений, ориентированных на конкретный бизнес-процесс.

EBIS — это предоставление набора инструментов BI, которые ранее поставлялись в виде разрозненных продуктов. Корпоративные BI-наборы обладают большей масштабируемостью и могут распространяться не только на внутренних пользователей, но и на ключевых заказчиков, поставщиков и др. Из-за тесного родства Web и корпоративных BI-наборов некоторые поставщики описывают свои BI-наборы как BI-порталы. Типичные EBIS поставляют Business Objects и Cognos.

Платформы BI предлагают наборы инструментов для создания, внедрения, поддержки и сопровождения BI-приложений. Имеются насыщенные данными приложения с «заказными» интерфейсами конечного пользователя, организованные вокруг специфических бизнес-проблем, с целевым анализом и моделями. Платформы BI, хотя и не так быстро растут и широко используются как EBIS, являются важным сегментом благодаря ожидаемому и уже происходящему росту BI-приложений. На данный момент целые семейства продуктов СУБД обеспечивают возможности BI по умолчанию.

BI-приложения обычно ориентированы на конкретную функцию организации или задачу, например, анализ и прогноз продаж, финансовое бюджетирование, прогнозирование, анализ рисков, анализ тенденций, «churn analysis» (анализ оттока клиентов) в телекоммуникациях и т.п. Они могут применяться и более широко как в случае приложений управления эффективностью предприятия (Enterprise Perfomance Management) или системы сбалансированных показателей (Balanced Scorecard).

Кроме перечисленных инструментов, в состав BI могут входить пакеты статистического анализа, анализ временных рядов, оценки рисков, средства моделирования, пакеты для нейронных сетей, средства нечеткой логики и экспертные системы.

Следует отметить практически обязательные в настоящее время средства для графического оформления результатов в BI: средства деловой и научно-технической графики; «приборные доски», средства аналитической картографии и топологических карт; средства визуализации многомерных данных.

Архитектура BI-технологии состоит из двух слоев: инфраструктуры и прикладных сервисов (функциональности). Инфраструктурный слой включает информационные ресурсы, администрирование и сети. На этом слое данные собираются, интегрируются и становятся доступными. Хранилище данных является одним из возможных компонентов

инфраструктурного слоя. Прикладные сервисы включают все BI-сервисы, такие как механизмы запросов, анализа, генерации отчетов и визуализации, а также средства безопасности и метаданные.

При использовании BI снижается прямая зависимость от подразделений ИТ, изготавливающих по заказу отчеты или запросы. Становится возможен переход от статических регламентных отчетов к «живому отчету», аналитики получают возможность проводить кросс-тематический анализ и построение сводных отчетов с нуля, имея семантический слой, описывающий показатели и разрезы корпоративной информации. Эти же средства могут использовать программисты для быстрого создания регламентных, параметрических отчетов. Web-доступ к BI позволяет обеспечивать реальное корпоративное информационное пространство и коллективную работу сотрудников. Рассмотрим некоторые BI системы.

IBM Cognos. Данный продукт - один из лидеров сегмента рынка BI. Основные свойства данного комплекса можно рассматривать как основополагающие и единые для всего сегмента современных BI и СППР. Они требуют обязательного учета в выявлении свойств новых систем мониторинга и поддержки решений.

IBM Cognos – это единая платформа для анализа данных и построения отчетности, которая решает задачи измерения и управления эффективностью деятельности предприятия (Corporate Performance Management, CPM).

Целевые клиенты этой и подобных систем – это компании среднего и крупного бизнеса самых разных отраслей, в т.ч. территориально-распределенные холдинги. Некоторые клиенты Cognos: IKEA, Toyota, Nestle, Coca-Cola, ТНК ВР, ИФД «Капиталъ», МЭР, РОСНО и др.

Управление предприятием в IBM Cognos организовано по принципу замкнутого цикла: «Мониторинг» (IBM Cognos Metrics Studio), «Анализ» (IBM Cognos Analysis Studio) и «Планирование» (IBM Cognos Planning). Решение IBM Cognos позволяет ответить на вопросы: «Что происходит?», «В чем причина?» и «Что делать?».

Модуль Metrics Studio системы IBM Cognos BI позволяет управлять предприятием (в том числе в части управленческого учета, бюджетирования и финансового планирования) в соответствии с принципами Balanced Scorecard (BSC) и Key Performance Indicators (KPI). Система позволяет сформулировать стратегические и тактические планы развития предприятия в виде конкретных целевых показателей (как финансовых, так и нефинансовых), установить взаимосвязи между ними, а также отслеживать реализацию корпоративной стратегии.

IBM Cognos Analysis Studio предоставляет пользователю возможности для всестороннего анализа и представления данных. Источниками могут быть OLAP-кубы, либо многомерно смоделированные реляционные данные. Analysis Studio позволяет выявлять первопричины отклонений фактических показателей от плановых значений. Пользователь, в зависимости от

полномочий, имеет возможность анализировать большие массивы информации при мгновенном отклике системы.

IBM Cognos Report Studio - инструмент создания сложных профессиональных отчетов, позволяющий использовать данные из различных источников и представлять их как в виде классических отчетов, так и информационных панелей.

IBM Cognos Query Studio позволяет быстро и легко извлекать данные из разнородных источников (ERP-система, файлы Excel и т.д.) в режиме реального времени. Используя Интернет-браузер, пользователи могут управлять набором заранее определенных запросов, изменяя и модифицируя их для получения оперативных данных.

IBM Cognos Event Studio - компонент, позволяющий определять действия системы при наступлении определенных пользователем событий. Примером события может быть изменение некоторого показателя, превышающее определенный пользователем диапазон. Действием может быть отправка сообщения по электронной почте, запуск отчета, запуск административного процесса, запуск процесса, относящегося к системе планирования.

IBM Cognos Planning содержит набор инструментов для осуществления финансового планирования и бюджетирования в режиме реального времени и позволяет реализовать процесс бюджетирования как «снизу-вверх» так и «сверху-вниз». Система обладает возможностью обратного пересчета, которая позволяет изменять агрегированные показатели и отслеживать влияние таких изменений на связанные показатели нижнего уровня.

IBM Cognos Planning Administrator отвечает за реализацию процедур финансового планирования и бюджетирования на предприятии, консолидацию данных. Данный модуль предоставляет финансовым менеджерам инструменты для моделирования, прогнозирования и планирования, а также позволяет осуществлять обмен информацией между всеми участниками бюджетного процесса (в т.ч. без постоянного подключения к вычислительной сети). Встроенный математический аппарат позволяет решать задачи многокритериальной оптимизации.

IBM Cognos Planning Contributor User представляет собой инструмент сбора данных от всех участников процесса финансового планирования и бюджетирования. Работа организуется путем заполнения веб-шаблонов документов, представляющих собой многомерные кубы данных, настраиваемые под конкретных пользователей.

Технологии:

- интерфейс пользователя - пользователь может работать как через стандартный Windows-интерфейс, так и через Web (Интернет-браузер) и непосредственно через Microsoft Excel. Есть возможность автономной работы. Продукт локализован для использования в России и СНГ;

- СУБД - поддерживаются Ms SQL Server, Oracle, DB2, Informix, Sybase и др.;
- бизнес-логика системы является открытой для пользователя/разработчика. Есть возможность адаптировать систему к специфике бизнеса и самостоятельно развивать ее;
- возможность использования OLAP-решений сторонних разработчиков - Microsoft, SAP BW, Essbase, IBM DB2 OLAP.

Архитектура системы IBM Cognos позволяет масштабировать решение на десятки тысяч пользователей и эффективно обрабатывать данные в компаниях масштабов Fortune 1000.

Комплекс содержит штатные средства интеграции с внешними системами (бухгалтерскими, ERP, HR, MES и т.д.). Источниками данных для системы IBM Cognos могут быть не только базы данных и корпоративные приложения, а и слабоформализованные данные (например, хранящиеся в файлах Excel).

SPSS.«Statistical Package for the Social Sciences» - «статистический пакет для социальных наук», http://www.spss.com/.

Программный комплекс, предназначенный для статистической обработки данных. Один из лидеров рынка в области коммерческих статистических продуктов, предназначенных для проведения прикладных исследований в социальных науках.

Программные продукты SPSS позволяют оперативно получать аналитическую информацию, наглядно представлять результаты в виде таблиц и диаграмм, а также распространять и внедрять полученные результаты. Это позволяет находить ключевые факторы, взаимосвязи и тенденции в данных и своевременно принимать оптимальные решения.

SPSS Statistics - это модульный, интегрированный программный комплекс, охватывающий все этапы аналитического процесса: планирование, сбор данных, доступ к данным и управление данными, анализ, создание отчетов и распространение результатов. Интерфейс включает в себя все функции управления данными, статистические процедуры и средства создания отчетов для проведения анализа любой степени.

SPSS представляет комплексное решение в области статистического анализа и Data Mining, в т.ч. для государственных организаций. Вот некоторые способы применения программных продуктов и услуг SPSS в государственном управлении:

- выявление случаев мошенничества, злоупотреблений и нецелевого расходования средств;
- проведение опросов и обследований;
- безопасность.

Список программных продуктов/модулей SPSS:

- Amos - моделирование структурными уравнениями для поиска скрытых взаимосвязей в данных;
- SamplePower - определение размера выборки, соответствующего бюджету и решаемым задачам;
- SmartViewer for Windows - распространение отчетов, созданных в SPSS для Windows;
- SmartViewer Web Server - распространение в Интернете интерактивных отчетов, созданных в SPSS для Windows;
- SPSS Advanced Models - анализ сложных взаимосвязей в данных;
- SPSS Base - основа статистического пакета SPSS для Windows;
- SPSS Categories - прогнозирование категориальных данных;
- SPSS Classification Trees - позволяет непосредственно в SPSS для Windows строить деревья классификаций и решений, идентифицировать группы, находить взаимосвязи в данных и предсказывать будущие события;
- SPSS Complex Samples - разработка сложных выборок и учет сложности выборки в процессе анализа;
- SPSS Conjoint - определение предпочтений клиентов;
- SPSS Data Entry - инструменты для организации ввода и чистки данных;
- SPSS Exact Tests - точный анализ малых выборок и разрезов;
- SPSS Data Preparation - позволяет упрощать процесс проверки данных, устраняя интенсивную ручную проверку, и получать более точные результаты;
- SPSS Missing Value Analysis - корректная импутация пропущенных значений;
- SPSS Regression Models - модели прогнозирования;
- SPSS Server - анализ централизованно хранящихся данных большого объема;
- SPSS Tables - эффективное представление результатов;
- SPSS Trends - прогнозирование временных рядов;
- TextSmart - быстрый и полный анализ ответов на открытые вопросы опросов и обследований.

Современные OLAP системы включают в свой состав инструменты для расчета ключевых индикаторов эффективности – Key Performance Indicator (KPI). KPI определяются как оценка полезности значений числовых мер, используемых в OLAP обычно в шкале от -1 до 1. Шкала KPI разбивается на 3-5 интервалов и для каждого из интервалов устанавливается когнитивный образ оценки состояния (красный/желтый/зеленый светофор, «смайлы», progress bar и т.д.). Кроме того для KPI может быть задан вес и тогда несколько KPI можно агрегировать

в KPI более высокого уровня методом взвешенной суммы. KPI роднит OLAP системы с классическими СППР.

1.3.2. СППР оболочки

Среди СППР, созданных по принципу оболочек, можно выделить следующие системы, которые представлены на коммерческом рынке и успешно эксплуатируются [5, 32-33, 49, 92, 110, 164-187]:

- Expert Choice – система, созданная при участии видного американского ученого Т. Саати, автора метода анализа иерархий, и получившая широкое распространение. Expert Choice обладает развитым интерфейсом пользователя, работает только по методу анализа иерархий. Для попарного сравнения критериев широко используются диаграммы в виде кругов и столбцов. Результаты оценки альтернатив также представлены в виде диаграмм. Для нормировки показателей используются как линейные, так и квадратичные полиномы. В СППР Expert Choice 2000 существует возможность присоединения источника ODBC для доступа к реляционной БД. В БД хранятся оцениваемые альтернативы. ODBC поддерживает большинство современных СУБД и позволяет получить доступ к альтернативам практически в любой реляционной БД. Результаты оценки альтернатив также записываются в БД.
- Super Decisions - в основе системы лежит метод аналитических сетей (МАС) – обобщение метода анализа иерархий.
- Criterium DecisionPlus – это американская СППР, которая позволяет работать как по методу анализа иерархий, так и с использованием многомерной теории полезности. Интерфейс программы во многом похож на интерфейс Expert Choice. Существенным недостатком Criterium DecisionPlus является отсутствие многопользовательского режима работы. Открытость Criterium DecisionPlus заключается в наличии интерфейса автоматизации для управления СППР извне. Таким образом, можно пристыковывать СППР к внешним программам. Обратный подход, когда СППР пристыковывает к себе внешний модуль, не реализован.
- Decision Lens - в основе системы лежит метод анализа иерархий.
- ИАС “ОЦЕНКА и ВЫБОР” – разработка отечественных ученых Д.А. Абдрахимова и А.И. Иоффина применяет метод взвешенной суммы и идеальную точку, позволяет строить иерархию показателей и визуализировать оцениваемые решения.
- «Быстрый прототип» - работает в среде MS Excel, позволяет учитывать мнение нескольких экспертов. Проводит ранжирование альтернатив с помощью метода анализа иерархий. Работает только в среде MS Excel. Открытость СППР «Быстрый

прототип» достигается за счет доступности всего исходного текста, написанного на Visual Basic for Application. К сожалению, интерфейс пользователя СППР «Быстрый прототип» трудно осваивать.

- DASS – отечественная разработка, использующая метод качественного учета важностей по отдельным компонентам векторного критерия В.В. Подиновского.
- FEASIBLE GOALS – программное обеспечение решения линейных многокритериальных задач оптимизации использует метод А.В. Лотова для визуализации границы Парето.
- ПК MOVI – программный комплекс, используемый для постановки и решения задач многокритериальной оптимизации в интерактивном режиме. Использует методику Р.Б. Статникова.
- Web-HIPRE – система предназначенная для решения многокритериальных задач методами анализа иерархий и теории полезности, разработана в Финляндии, ориентирована на работу в сети Интернет.

Одними из первых СППР, основанных на оценке полезности, можно считать программные комплексы, разработанные в Рижском политехническом институте под руководством проф. А.Н. Борисова, а именно:

- программная система DELTA, предназначенная для выбора многокритериальных альтернатив при нечетких ограничениях с использованием методов теории полезности;
- диалоговая система SDMIS (Simulation and Decision Making Interactive System), предназначенная для многокритериальной оценки и выбора альтернатив при нечеткой исходной информации относительно характеристик альтернатив и предпочтений ЛПР;
- диалоговая система "Альтернатива", предназначенная для решения задач поддержки и принятия решений в условиях риска или неопределенности при наличии нескольких критериев.

Как показал проведенный анализ, автоматизированная поддержка решений в многокритериальных задачах разрабатывается достаточно активно. Однако можно указать на ряд общих, характерных для многих систем, недостатков. Большинство СППР реализует только один метод свертки векторного критерия и, как правило, это метод взвешенного суммирования, о минусах которого уже сказано выше. Большинство СППР не может работать с лексическими показателями. Для них требуется предварительная оцифровка, т.е. перевод в количественную шкалу. Однако для ЛПР затруднительно высказывать количественные суждения о лексических переменных. Большинство СППР не позволяют учитывать зависимость показателей по предпочтениям. Большинство СППР не являются открытыми, не имеют специфицированной структуры базы данных, что затрудняет интеграцию их с внешними приложениями для

получения данных об оцениваемых альтернативах и выгрузки результатов поиска рационального решения. Кроме того, достаточно редко встречаются системы, ориентированные на Web технологии.

В таблицах 1.1 и 1.2 приведены некоторые характеристики СППР, представленных на коммерческом рынке. К сожалению, далеко не по всем системам есть полная информация, поэтому между таблицами нет полного пересечения.

Табл. 1.1. Основные характеристики СППР.

Название	Метод парных сравнений	Временной анализ	Анализ чуствитель-ности	Групповые оценки	Анализ рисков	Web-интер-фейс
1000Minds	Да	Нет	Да	Да	Нет	Да
Analytica	Нет	Да	Да	Нет	Да	Да
Criterium DecisionPlus	Нет	Нет	Да	Нет	Нет	Нет
Decision Lab	Да	Нет	Да	Нет	Да	Нет
Decision Manager	Нет	Нет	Да	Нет	Да	Нет
DPL Syncopation	Нет	Нет	Да	Нет	Да	Нет
DSS/UTES	Да	Нет	Да	Да	Да	Нет
ERGO	Да	Нет	Да	Нет	Нет	Да
Expert Choice	Да	Нет	Да	Да	Да	Да
Hiview3	Нет	Нет	Да	Да	Нет	Нет
Logical Decisions	Нет	Нет	Да	Нет	Нет	Нет
MindDecider	Да	Да	Да	Да	Да	Нет
RPM-Decisions	Нет	Нет	Да	Нет	Нет	Да
TreeAge Pro	Нет	Нет	Да	Нет	Да	Нет
Vanguard Studio	Нет	Нет	Да	Да	Да	Да

Табл. 1.2. Дополнительная информация по некоторым версиям ПО.

Название	Адрес в интернете	Стои-мость	Методы	Технология	Платформа	Интеграция
TreeAge Pro Suite (Base + Excel + Медицинский модуль)	http://www.treeage.com/	1665 (полная) 575 (на год)	деревья решений, марковские модели, диаграммы влияния		Windows (Vista, XP, 2000, 98)	
Expert Choice Comparion Core	http://www.expertchoice.com/		Метод анализа иерархий	.net	*Windows 2003 server*	Свои сайты Oracle MS Project

Название	Адрес в интернете	Стои-мость	Методы	Технология	Платформа	Интеграция
Super Decisions	http://www.superdecisions.com	Бесплатно, бета	Метод аналити-ческих сетей	**Tcl/Tk**	Windows MacIntosh Linux Solaris NetBSD	
Decision Lens Web	http://www.decisionlens.com/		Метод анализа иерархий	Flash		
DecisionPad 3	http://www.apian.com/	$845 полная $49 на месяц	Метод анализа иерархий, полезности, QFD		Windows	
DSS/UTES	http://dss-utes.sourceforge.net/		Предпочтения, метод анализа иерархий, метод идеальной точки, медиана Кемени	C++	Windows	MS Excel, COM, OLE
Decision Supporter	http://decisionsupporter.com/		Метод анализа иерархий	asp		

1.4. СППР DSS/UTES

СППР DSS-UTES предназначена для поддержки многокритериальных решений в сложной информационной среде. Аббревиатура DSS-UTES (Decision Support System – UTility Estimator) переводится как система поддержки решений – выявления предпочтений. Оригинальность системы заключается в процедуре выявления предпочтений ЛПР. При этом предполагается, что, рассматривая значения двух совокупностей показателей, он всегда может сказать, какая из них, с его точки зрения, предпочтительнее, или же они равноценны. Весьма существенно, что эта процедура не требует от него количественных оценок степени предпочтений, что всегда затруднительно, достаточно лишь сопоставления. В системе DSS-UTES такие сопоставления выполняются на плоских (двумерных) сечениях функции предпочтений [22, 29].

1.4.1. Основные функции DSS/UTES

Основные функции DSS/UTES и их взаимосвязи отражены на схеме, приведенной на рисунке 1.2.

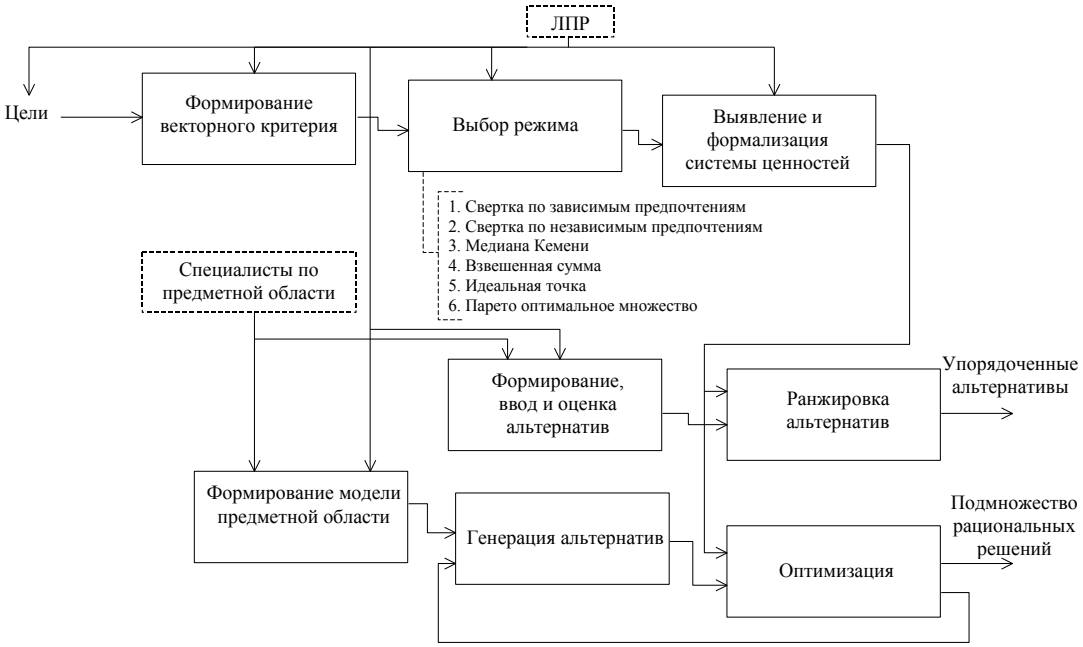


Рис. 1.2. Функциональная структура СППР DSS/UTES.

При выделении функций было принято ещё одно принципиальное положение: СППР должна быть по возможности универсальной, в том смысле, что пользователь имеет возможность по своему усмотрению выбирать режимы работы системы, реализующие методы скаляризации векторного критерия. Эти методы, подробно описанные выше, разнятся между собой сложностью их воплощения и адекватностью отображения системы ценностей пользователя.

Кратко функции СППР, приведенные на рисунке 1.2., заключаются в следующем:

Формирование векторного критерия. Работа пользователя с СППР начинается с определения цели и формализации её в виде векторного критерия. В работе [29] формулируются желательные свойства, которыми должен обладать векторный критерий, но они носят рекомендательный характер и не поддаются формализации настолько, чтобы проверку этих свойств можно было встроить в СППР; пока набор этих свойств можно вставлять только как подсказку пользователю при редактировании. К желательным свойствам векторного критерия относятся:

- полнота – набор критериев должен охватывать все существенные аспекты решаемой задачи;
- действенность – набор критериев может быть с пользой применен при анализе задачи;

- разложимость – набор критериев можно разбить на части, чтобы упростить решаемую задачу;
- неизбыточность – критерии не должны дублировать одни и те же аспекты решаемой задачи;
- минимальность – критерий должен иметь по возможности минимальную мерность;

Для каждого компонента векторного критерия задаются ряд атрибутов:

- имя критерия;
- тип (лексический или численный);
- шкала;
- единицы измерения;
- неформальное текстовое описание;
- минимальное и максимальное значение;
- направление улучшения (на минимум или на максимум) или идеальное значение.

Выбор режима. После формирования векторного критерия, необходимо указать режим работы СППР, т.е. какой метод свертки векторного критерия будет использоваться. В разрабатываемой СППР реализованы следующие режимы:

1-й режим – свертка по зависимым предпочтениям.

Зависимость по предпочтениям означает, что отношения предпочтения двух произвольно взятых показателей зависят от того, какие значения приняли другие показатели. Это наиболее общий и, как правило, наиболее часто встречающийся случай.

Пользователь вводит свои предпочтения на полном множестве альтернатив (на всех возможных комбинациях значений свертываемых критериев), не задумываясь о том, реализуемы они или нет. Если решения не достижимы, то это определится в ходе решения задачи и модель просто отбросит их. Определение предпочтений на полном множестве альтернатив дает возможность использовать систему при любых модельных ограничениях.

Детально процедура выявления и формализации предпочтений будет описана в следующем разделе.

2-й режим – свертка по независимым предпочтениям.

В этом режиме отношение предпочтений каждой пары показателей не зависит от значений остальных. Если пользователь выбирает такой режим, ему необходимо не сравнивать между собой различные показатели, а оценить по предпочтениям значения каждого показателя. Далее для скаляризации применяется один из формальных методов свертки, что вполне допустимо, т.к. зависимость по предпочтениям предполагается отсутствующей. Этот режим очень прост и применим для критериальных пространств любой мерности.

3-й режим - среднего ранжирования (Медиана Кемени).

В этом режиме вначале определяются ранжировки альтернатив по каждому из показателей, а затем на этой основе вычисляется среднее ранжирование с использованием медианы Кемени. Режим среднего ранжирования удобен тем, что не требует от пользователя никаких специальных действий и выполняется практически мгновенно. Он применим, если пользователь полагает все показатели равно важными и независимыми по предпочтениям.

4-й режим – взвешенная сумма.

В этом режиме предлагается осуществлять свертку векторного критерия на основе использования взвешенной суммы. Для назначения важностей локальных критериев предусмотрено использование процедуры парных сравнений или непосредственное их назначение.

5-й режим – идеальная точка.

Идеальная точка определяется наилучшими значениями по каждому из показателей. Для вычисления расстояния до идеальной точки используется евклидова метрика.

6-й режим – Парето оптимальность.

В данном режиме предусмотрена процедура выявления недоминируемых альтернатив с использованием принципа Парето.

Как видно из описания режимов, разрабатываемая СППР по своим возможностям (количеству используемых методов свертки) превосходит все аналогичные, представленные на коммерческом рынке, а благодаря наличию первого режима, является уникальной.

Выявление и формализация системы ценностей. В критериальном пространстве определяются предпочтения на полном множестве альтернатив (если выбран 1-й режим) или предпочтения на значениях каждого показателя (если выбран 2-й режим) или определяются веса показателей (если выбран 3-й режим) и т.д. Часто выявление и формализация системы ценностей наиболее долгий этап работы ЛПР с системой. Эта процедура выполняется один раз при настройке системы.

Формирование, ввод и оценка альтернатив. Данная процедура выполняется пользователем совместно с экспертами и специалистами по предметной области.

Ранжировка альтернатив выполняется автоматически. В заключении процедуры ранжировки СППР выдает список альтернатив, упорядоченный по убыванию предпочтительности (причем, ранжировка не всегда строгая) или список только первых n наиболее предпочтительных альтернатив.

Формирование модели предметной области. В случае необходимости осуществления параметрической и(или) структурной оптимизации пользователь формирует модель, которую затем необходимо подключить к СППР. Входом этой модели являются значения оптимизируемых параметров, а выходом значения локальных критериев.

Генерация альтернатив. СППР подает на вход модели предметной области значения оптимизируемых параметров и получает соответствующие им значения локальных критериев.

Оптимизация. По значениям локальных критериев система определяет уровень предпочтительности альтернативы и использует его в качестве целевой функции (для первого и второго режимов). В режимах взвешенной суммы и идеальной точки целевой функцией служит скаляризованное значение критерия. Третий и шестой режимы в задачах оптимизации не используются. Если условия оптимальности выполнены, то СППР выдает оптимальное решение, а так же несколько наиболее близких к нему, на множестве которых ЛПР осуществляет выбор.

1.4.2. Средства выявления и формализации предпочтений пользователя

Рассмотрим наиболее принципиальный режим: свертка по зависимым предпочтениям. При этом предполагается, что, рассматривая значения двух совокупностей показателей, пользователь всегда может сказать, какая из них, с его точки зрения, предпочтительнее или же они равноценны. Весьма существенно, что эта процедура не требует от него количественных оценок степени предпочтений, что всегда затруднительно, достаточно лишь сопоставления. В системе DSS/UTES такие сопоставления выполняются на плоских (двумерных) сечениях функции предпочтений. Это делается следующим образом.

Пусть, имеется:

n – критериев оценки альтернатив;

m_j – число градаций шкалы j–го критерия (каждый из критериев имеет шкалу, причем, как минимум, порядковую);

$X_j = \{x_{j,1}, x_{j,2}, \ldots, x_{j,mj}\}$ – оценки на шкале j-го критерия (градации);

n – мерный гиперкуб, каждый j-й критерий образует соответствующую сторону (измерение) гиперкуба;

$X_1 \otimes X_2 \otimes \ldots \otimes X_n$ – сетка, построенная в гиперкубе, каждая точка которой является одной из возможных комбинацией значений по критериям.

Из множества введенных показателей пользователь выбирает любые два (например, 1 и 2), которые условимся называть активными. Желательно, но не обязательно, чтобы они обладали наибольшим числом градаций или же представлялись ему наиболее важными. Оставшиеся n-2 показателя – пассивные.

Активные показатели образуют решетку, содержащую $m_1 * m_2$ клеток. Каждая клетка отражает определенную комбинацию значений активных показателей. Пользователь должен расставить свои предпочтения на этой решетке. Для этого ему надо раскрасить клетки в различные оттенки одного цвета, который он может выбрать по своему вкусу. В системе по

умолчанию принята черно-белая цветовая шкала. Принято также, что клетка, имеющая более светлый оттенок, предпочтительнее, чем та, что темнее. Соответственно клетки одной тональности - равноценны. Перед раскрашиванием плоскости активных критериев пользователь должен назначить значения всех пассивных показателей, при которых он будет производить раскраску. Эти значения он выбирает по своему усмотрению, но выполнение дальнейших процедур упростится, если он каждому пассивному критерию припишет его минимально допустимое значение. По окончании раскрашивания пользователь изменяет значение одного из пассивных показателей и производит раскраску снова. И такая процедура должна повторяться до тех пор, пока не будут определены предпочтения на всех двумерных сечениях, соответствующих всем возможным комбинациям значений пассивных показателей. Если пользователь осуществляет ввод предпочтений при фиксированных показателях с номерами 1 и 2, то число сечений, на которых он должен определить предпочтения, определяется соотношением:

$$\prod_{j=3}^{n} m_j, \text{ если } m_j = m = \text{const}, \text{ то } \prod_{j=3}^{n} m_j = m^{(n-2)}$$

Число сечений может быть велико. Уже для 3-х фиксированных показателей с 4-я градациями оно составляет $4^3=64$.

В результате в базу системы будет записана информация в виде, позволяющем сравнить по предпочтениям пользователя две любые совокупности значений показателей. Для использования этой информации в задачах оптимизации и ранжирования ей необходимо придать числовую форму. Достигается это путем построения ФП. При этом предполагается, что все компоненты векторного критерия дискретны. Если они не дискретны, то их предварительно необходимо оцифровать в дискретной шкале.

При раскраске двумерных сечений для облегчения работы пользователя на экран выводится палитра полутонов, и работа пользователя при назначениях предпочтений сводится к выбору из этой палитры нужного полутона. Эти полутона расположены в порядке возрастания "светлости", и их можно перенумеровать, например, от черного до белого. Тогда для сопоставления совокупностей значений показателей достаточно сравнить номера присвоенных им полутонов. Та, чей номер выше, - предпочтительнее.

На рисунке 1.3. приведена экранная форма, соответствующая этапу формирования функции предпочтений. Показаны раскрашенное двумерное сечение ФП, перечень пассивных показателей с указанием "замороженных" значений и палитра полутонов. Соответственно, чем больше полутонов содержит палитра, тем выше разрешающая способность системы по предпочтениям. В частности, для того, чтобы различать по предпочтениям все возможные комбинации значений показателей, их число должно равняться числу полутонов палитры. Но

практически требуемое (или используемое) количество полутонов заметно меньше, т.к. существует большое число равных по предпочтениям комбинаций значений показателей. В частности, равными по предпочтениям являются все комбинации, в которых хотя бы один показатель имеет недопустимое значение.

Описанная процедура выявления и формализации предпочтений выполняется относительно просто, если число введенных показателей не превышает 5-ти и количество градаций каждого не слишком велико. В дальнейшем ее трудоемкость возрастает настолько резко, что она может стать практически невыполнимой. Это, в первую очередь, связано с тем, что при переходе от одного пассивного показателя к другому и при изменении их "замороженных" значений приходится запоминать предыдущие назначения.

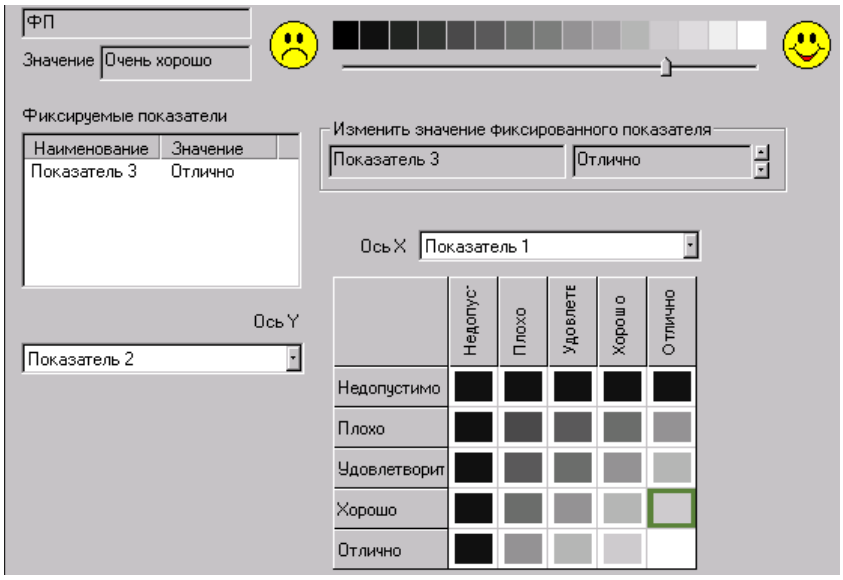


Рис. 1.3. Окно редактирования предпочтений.

Для снятия этого ограничения предлагается комплекс мер технического и методологического характера, в частности: доопределение ФП, агрегирование ФП и построение ФП для случая показателей, независимых по предпочтениям.

1.4.3. Архитектура DSS/UTES

Трехуровневая архитектура СППР DSS/UTES приведена на рисунке 1.4. Наиболее принципиальным является структура промежуточного уровня, представленного набором компонент COM [74,75]. В основе COM лежат два понятия: компонент и интерфейс. Компонент - это объект, содержащий определенный набор данных и процедур для их обработки. Интерфейс представляет собой описание определенного набора процедур. Компонент может поддерживать несколько интерфейсов. Один и тот же интерфейс может поддерживаться различными компонентами. Различные компоненты, поддерживающие одинаковые интерфейсы, принадлежат к одной категории. Внутренняя реализация компонент может быть

различной, они могут быть созданы в различных системах программирования. Поэтому данная архитектура является методологической основой для построения различных систем поддержки решений, работающих по предпочтениям пользователя.

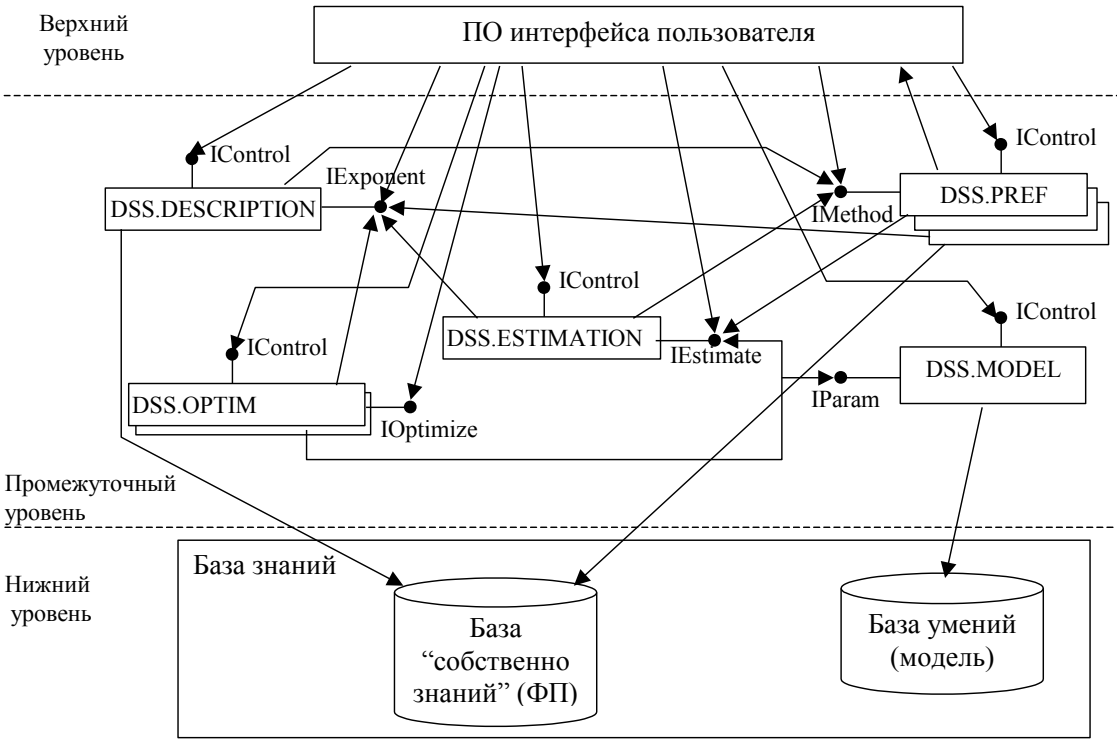


Рис. 1.4. Трехуровневая архитектура ПО СППР DSS/UTES.

Рассмотрим описание основных компонентов, их категорий и интерфейсов.

Компонент DSS.DESCRIPTION предназначен для доступа к атрибутам векторного критерия, определения режима работы, проверки на возможность применения выбранного режима работы, сохранения описания векторного критерия в базе знаний.

Категория компонентов свертки векторного критерия DSS.PREF. В данную категорию может входить произвольное количество компонентов, реализующих доступ к методам свертки векторного критерия. Важно то, что регистрация новых методов свертки происходит на уровне настроек.

Компонент DSS.ESTIMATION отвечает за ввод альтернатив, их ранжирование посредством запуска компонент из DSS.PREF, расчет значения целевой функции при решении задачи оптимизации.

Категория компонентов, реализующих модели предметных областей DSS.MODEL. Они позволяют задавать значения параметров модели и по ним получать значения локальных критериев. Простота интерфейсов модели предметной области дает возможность реализовывать их разработчикам, незнакомым с архитектурой СППР.

Категория компонентов оптимизации DSS.OPTIM позволяет реализовать доступ к процедуре оптимизации. Компонент оптимизации должен обеспечить выбор модели

предметной области из категории DSS.MODEL. Процедура оптимизации выглядит следующим образом:

1. Пользователем определяются начальные значения оптимизируемых параметров, если это необходимо выбранному методу оптимизации.
2. Компонент оптимизации вызывает компонент модели предметной области для определения значений локальных критериев по значениям оптимизируемых параметров. Если поданные на вход модели значения параметров являются недопустимыми, то компонент модели предметной области возвращает признак недопустимости и осуществляется переход на шаг 5.
3. Компонент оптимизации вызывает компонент DSS.ESTIMATION для получения значения целевой функции по значениям локальных критериев.
4. Компонент оптимизации проверяет выполнение условий на нахождение оптимального значения целевой функции. Если оптимум найден, то следует переход на шаг 6.
5. Компонент оптимизации генерирует новые значения оптимизируемых параметров (т.е. генерация альтернатив), после чего осуществляется переход на шаг 2.
6. Компонент оптимизации формирует подмножество из n рациональных решений, включающих оптимальное и близкие к нему решения.

Детали реализации п.п. 4 и 5 зависят от выбранного метода оптимизации.

Интерфейс IControl служит для управления любым компонентом промежуточного уровня, его сохранением, настройкой компонента, диагностики ошибок.

Интерфейс IExponent должен предоставлять возможности для управления формированием описания векторного критерия и для выбора режима свертки критерия.

Интерфейс IEstimate должен позволять ранжировать альтернативы, вычислять оценки альтернатив.

Интерфейс IMethod используется для редактирования предпочтений, и сообщения методу свертки об изменениях критериального пространства.

Интерфейс IParam позволяет для модели предметной области по значениям параметров модели вычислять значение векторного критерия.

Интерфейс IOptimize позволяет выполнять оптимизацию на модели предметной области.

Основные модули СППР DSS/UTES написаны на языке C++ и реализованы как COM компоненты. COM позволяет реализовать ряд компонентов, применяемых для свертки векторного критерия различными способами и управляемых через типизированный интерфейс [38, 153, 159]. Например, метод парных сравнений критериев был написан на Delphi и подключен через COM интерфейс. Ввод альтернатив и вывод результатов ранжирования

альтернатив реализован на Visual Basic for Application в среде MS Excel через интерфейс OLE-автоматизации [153]. Система DSS/UTES 2.0 представляет собой законченный программный продукт, представленный на коммерческий рынок СППР. Программное обеспечение системы полностью является авторской разработкой [21].

1.4.4. Решение задачи оценивания районов размещения ракет в DSS/UTES

СППР DSS/UTES апробировалась в задаче многокритериального оценивания районов размещения ракет. На рисунке 1.5 показан интерфейс DSS/UTES в режиме формирования иерархии показателей. В примере, заданном заказчиком, все исходные показатели носят числовой характер, и при расчетах применен метод взвешенного суммирования с равными коэффициентами важности, что не исключает использование и основного режима DSS/UTES, учитывающего зависимость показателей по предпочтениям.

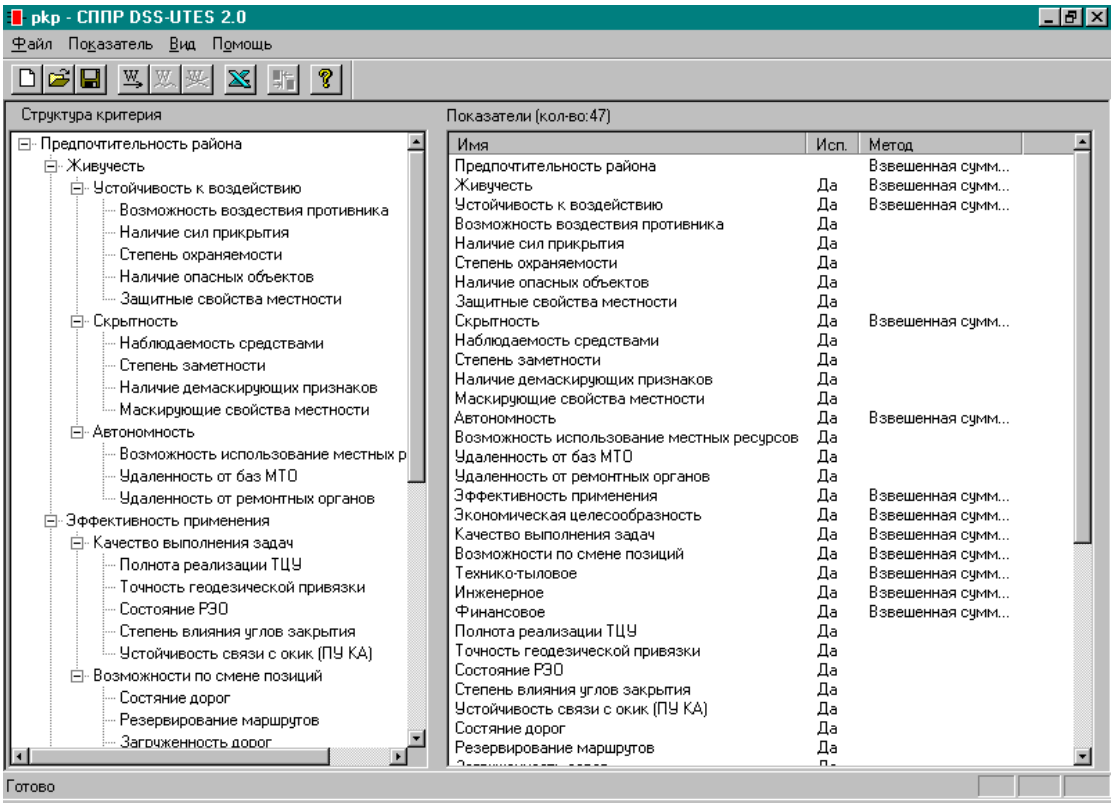


Рис. 1.5. Дерево оценки районов размещения ракет.

Рассмотренная задача выбора полевых позиционных районов размещения ракет является высокоразмерной, так как исходных показателей порядка 35. Для ее решения применялась процедура агрегирования показателей в группы. Так как в примере, заданном заказчиком, все исходные показатели носят числовой характер, то при расчетах применен метод взвешенного суммирования с равными коэффициентами важности, что не исключает использование и основного режима DSS/UTES учитывающего зависимость показателей по предпочтениям. Результаты расчета для четырех тестовых позиционных районов приведены в таблице 1.3.

Табл. 1.3.

Оценка районов

	Первый	Четвертый	Второй	Третий
Возможность воздействия противника	2	8	4	6
Наличие сил прикрытия	1	4	3	2
Степень охраняемости	2	5	5	3
Наличие опасных объектов	4	1	2	3
Защитные свойства местности	4	2	3	2
Наблюдаемость средствами	1	6	2	4
Степень заметности	4	2	2	3
Наличие демаскирующих признаков	2	6	3	4
Маскирующие свойства местности	5	2	4	3
Возможность использование местных ресурсов	2	4	3	2
Удаленность от баз МТО	4	1	2	3
Удаленность от ремонтных органов	4	1	2	3
Полнота реализации ТЦУ	4	3	2	3
Точность геодезической привязки	1	3	1	2
Состояние РЭО	2	5	4	3
Степень влияния углов закрытия	2	3	3	2
Устойчивость связи с ОКИК (ПУ КА)	2	5	3	4
Состояние дорог	2	5	3	4
Резервирование маршрутов	1	3	1	2
Загруженность дорог	1	4	3	2
Затраты ресурсов на маневр	4	1	3	2
Влияние сезонных изменения погоды	3	1	2	2
Затраты моторесурсов	7	2	4	4
Расход ГСМ	7	1	4	4
Затраты на поддержание автономности	6	1	3	3
Затраты на обеспечение жизнедеятельности	5	1	3	3
Затраты на инженерную разведку	5	1	2	2
Затраты на оборудование ПРП	5	2	2	2
Затраты на поддержание в готовности ПРП	6	2	3	3
Затраты на подготовку и поддержание маршрутов	6	2	3	3
Затраты на аренду земли	1	4	2	2
Затраты на возмещение ущербы среде	2	4	3	3
Оплата работ и услуг	2	4	3	3
Затраты на подготовку личного состава	4	1	2	2
Временные затраты на маневр	3	1	3	2
Итоговая предпочтительность:	3,17021753	3,01853329	2,94371788	2,75246323
Предпочтительность по живучести:	2,5453748	3,64819245	3,47004253	2,75738969
Предпочтительность эффективности применения:	2,19444444	3,27777778	2,44444444	2,58333333

	Первый	Четвертый	Второй	Третий
Предпочтительность экономической целесообразности:	4,77083333	2,12962963	2,91666667	2,91666667

1.5. Выводы по первой главе

Существует большое число методов решения многокритериальных задач. Они различаются:

- подходами, с помощью которых задача сводится к однокритериальной (скаляризация векторного критерия) или определяется отношение доминирования альтернатив;
- различной степенью участия ЛПР в их применении и возможностями адекватного учета его предпочтений;
- различными возможностями работы с разнородными компонентами векторного критерия (числовыми и лексическими);
- различными возможностями применения и вычислительной сложностью при возрастании размерности векторного критерия;
- достоинствами и недостатками, зависящими от условий применения метода.

Все методы имеют право на жизнь и выбор метода зависит от предпочтений ЛПР. Но, как показал анализ, наибольшими возможностями в адекватном учете мнения ЛПР и эффективности его применения, является метод основанный на функции предпочтений (ФП) по сечениям. Он так же является наиболее эффективным по быстроте оценки и ранжирования альтернатив, когда ФП уже построена. Серьезным недостатком этого метода является большой объем необходимой аналитической работы ЛПР в процессе построения ФП.

С точки зрения построения методологии унифицированной разработки СППР наибольший интерес представляют СППР построенные по принципу оболочек. Но многие имеющиеся образцы таких СППР обладают целым рядом недостатков:

- они, как правило, реализуют какой-либо один метод и в них отсутствует возможность применения различных методов;
- возможность их настройки на решение какой-либо конкретной многокритериальной задачи связана с большими дополнительными работами по созданию математического, алгоритмического и информационного обеспечения, реализующего такую настройку.

Наибольшими возможностями, которые могут быть использованы при создании методологии унифицированной разработки, обладает СППР DSS/UTES: возможностями использования различных методов, возможностями работы с разнородными показателями, возможностями использования наиболее адекватного учета мнения ЛПР в виде ФП,

возможностями интеграции модулей для реализации новых методов и подключения источников с альтернативами. Однако и DSS/UTES свойственные определенные недостатки:

- трудоемкая процедура построения ФП по сечениям;
- средства реализации возможностей расширения на базе COM-компонент ориентированы только на платформу Windows.

Глава 2. Практика разработки систем поддержки решений в задачах с высокой размерностью векторного критерия

2.1. «Автоматизированная система мониторинга муниципальных образований (АСМ МО)»

Целью создания «Автоматизированной системы мониторинга муниципальных образований (АСМ МО)» было совершенствование управления сферой социально-экономического развития муниципальных образований (МО) на основе разработки и внедрения информационных технологий в процесс социально-экономического мониторинга МО [29-31, 122].

Практика мониторинга и управления МО показала необходимость унификации показателей качества их функционирования [45]. При этом такие показатели не должны быть излишне "жесткими", чтобы обеспечивать учет индивидуальных особенностей МО, обусловленных их территориальным расположением, преимущественным характером деятельности, местными социальными и культурными особенностями и т.п. В то же время совокупность основных показателей должна обеспечивать возможность решения задач не только мониторинга, но и сопоставления деятельности различных МО с целью динамичного управления ими.

В развитие этого положения, Институтом Государственного Управления и Кадровой Политики РЭА им. Г. В. Плеханова была разработана "Система Основных Показателей для анализа и прогнозирования социально-экономического развития муниципальных образований (СОП)". Эта система включает в себя 277 исходных показателей (для одного МО), часть которых в дальнейшем агрегируется в обобщенные показатели более высокого (относительно исходных) уровня.

Последовательное многоуровневое агрегирование вначале исходных, а затем и обобщенных показателей, приводящее к сокращению их количества, сводит оценку деятельности МО к вычислению значений сравнительно небольшой группы показателей. Эти конечные показатели можно понимать как компоненты векторного критерия качества функционирования МО в целом. В СОП их четыре:

- ресурсы МО,
- муниципальное хозяйство,
- уровень жизни населения,
- социально-экономическое состояние и развитие МО.

Кроме того СОП содержит обобщенные показатели деятельности МО как по различным

направлениям, так и по уровням иерархии внутри этих направлений, что создает достаточно полную картину состояния МО.

Структурно СОП представляет собой многоуровневую систему взаимосвязанных величин, которая по отношению включения может рассматриваться как иерархическая.

Если обратиться ко всей совокупности социально-экономических образований страны, в которой МО выступают уже в качестве объектов, то структура оценок ее деятельности также носит характер иерархии, имеющей, как минимум, три уровня: муниципальный, региональный, федеральный. Таким образом задачи решаемые АСМ МО как на местных, так и на федеральном уровнях можно рассматривать как задачу мониторинга, оценки и управления иерархическими структурами. Такое обобщение позволит в дальнейшем более широко использовать разрабатываемые подходы и методы.

По согласованию с Миннацем в основу федеральной АСМ МО была положена структура СОП. Как следствие этого, разработка АСМ МО может быть формализована как адаптация АСМ иерархическими структурами к СОП. Однако это отнюдь не означает, что оценка деятельности МО должна ограничиваться содержанием и структурой СОП. АСМ МО является открытой как по отношению к содержательному наполнению, так и по отношению к применяемым системам программирования.

Основные функции, выполняемые АСМ МО, можно определить следующим образом -

На уровне отдельного МО:

- хранение исходной информации о состоянии МО (о ресурсах, о промышленном, культурном и прочих потенциалах и многое другое);
- вычисление различных показателей, характеризующих состояние МО;
- отбор информации для отображения в виде различных таблиц, графиков и диаграмм;
- определение значений качественных показателей на основе предпочтений пользователя системы;
- использование для оценки состояния МО предпочтений, задаваемых как сами работниками МО, так и работниками областного и федерального уровня;
- анализ динамики изменения показателей во времени. Определение причин, повлекших ухудшение или улучшение состояния МО;
- определение показателей, изменение которых в первую очередь необходимо;
- оценка различных управленческих решений с точки зрения изменения состояния МО.

На региональном и федеральном уровне:

- хранение информации о состоянии различных МО. Прием информации из баз данных отдельных МО;

- сравнение МО по отдельным показателям. Поиск МО с наилучшим (наихудшим) значением показателя. Упорядочивание МО по ухудшению или улучшению выбранного показателя. Табличное представление результатов для неформального анализа;
- сравнение МО по обобщенным показателям (например, качество жизни); для сравнения назначаются предпочтения пользователя.

Система проектируется так, чтобы для работы с нею пользователю было достаточно обладать минимальным опытом общения с компьютером, т.е. базовыми навыками работы в операционной системе Windows 95 или выше.

Наиболее принципиальным моментом разработки является создание логического программного интерфейса между базой данных АСМ МО и DSS/UTES, обеспечивающего загрузку информации о показателях МО из БД в DSS/UTES и занесение в БД из DSS/UTES вычисленных значений обобщенных показателей. Эти вопросы рассматриваются ниже.

2.1.1. Вычисление обобщенных показателей

На рисунке 2.1. приведено дерево показателей состояния МО, построенное в соответствии с СОП. Исходные показатели обозначены курсивом, а обобщенные – жирным шрифтом.

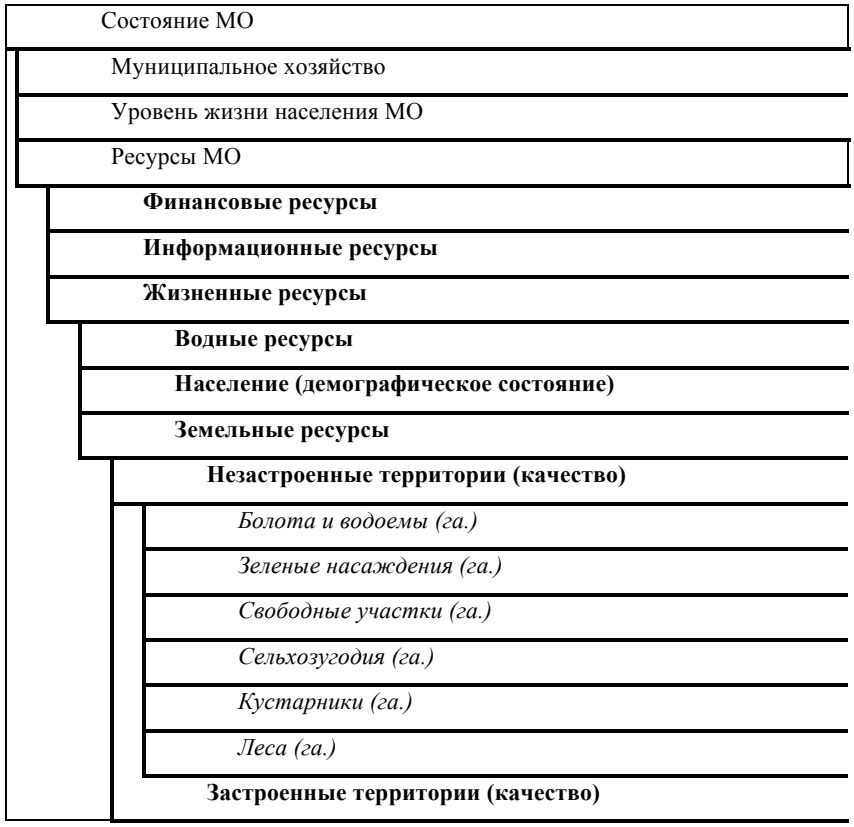


Рис. 2.1. Дерево показателей МО.

На рисунке 2.1. только одна из ветвей дерева показана полностью: от исходных показателей, сворачиваемых в обобщенный показатель низшего уровня «незастроенные территории», до показателя высшего уровня иерархии «Ресурсы МО». Обобщенные показатели других ветвей не раскрыты.

Из рисунка 2.1. видно, что показатели, агрегируемые в один обобщенный критерий, различаются в отношении типа и шкалы. Они могут быть полностью однородными как, например, в показателе «незастроенные территории», или же различаться как типами, так и шкалами, как, например, в показателе «ресурсы МО». Формальные и содержательные различия обобщенных показателей приводят к тому, что для каждого из них должен выбираться свой, наиболее ему соответствующий метод свертки (скаляризации). При выборе необходимо учитывать различие типов показателей (числовые и лексические) и шкал, наличие или отсутствие зависимости по предпочтениями и возможность учета системы ценностей ЛПР. Оболочка СППР DSS/UTES 2.0 отвечает указанным требованиям и может, как указывалось выше, использоваться как универсальный инструмент свертки векторного критерия или, в терминах АСМ МО, - вычисления обобщенного показателя. Поэтому было принято решение об интеграции DSS/UTES в структуру АСМ МО.

Необходимо заметить, что созданное в Академии им. Плеханова дерево показателей состояний МО длительное время оставалось абстрактной экономической структурой, не нашедшей практического применения, именно потому, что не было инструмента для свертки разнородных обобщенных показателей. Коротко опишем процедуру вычисления обобщенных показателей в АСМ МО.

На основе содержательного и формального анализов рассматриваемого обобщенного критерия выбираем метод его свертки. Эти методы были уже описаны выше.

Наиболее принципиальным и в то же время наиболее трудоемким является **режим свертки на основе зависимых предпочтений**. Поэтому начнем с его рассмотрения.

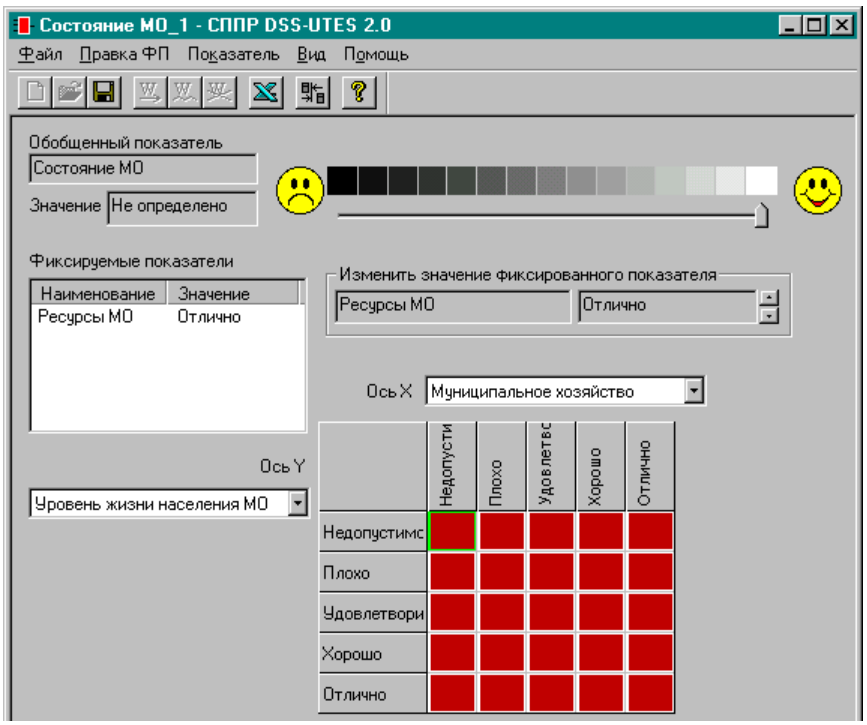


Рис. 2.2. Незаполненное основное окно DSS/UTES.

В примере иллюстрируется агрегирование трех обобщенных показателей:

- «Уровень жизни населения»;
- «Ресурсы МО»;
- «Муниципальное хозяйство»

в один обобщенный показатель более высокого уровня - **«Состояние МО».**

В окне на рисунке 2.2. изображено двумерное сечение функции предпочтений (ФП). Пользователь должен раскрасить его в черно-белые тона (палитра тонов приведена вверху справа). Чем предпочтительнее альтернатива (в данном примере это сочетание значений показателей «Уровень жизни населения» и «Муниципальное хозяйство» при заданном значении показателя «Ресурсы МО»), тем светлее должен быть прямоугольник. Если бы агрегировалось n показателей, (n-2) из них были бы заданы постоянными значениями.

На рисунке 2.3. показана «именованная» шкала предпочтений. Названия лексических термов пользователь может назначать по своему усмотрению. Важно лишь, что бы они отражали понятную ему последовательность предпочтений.

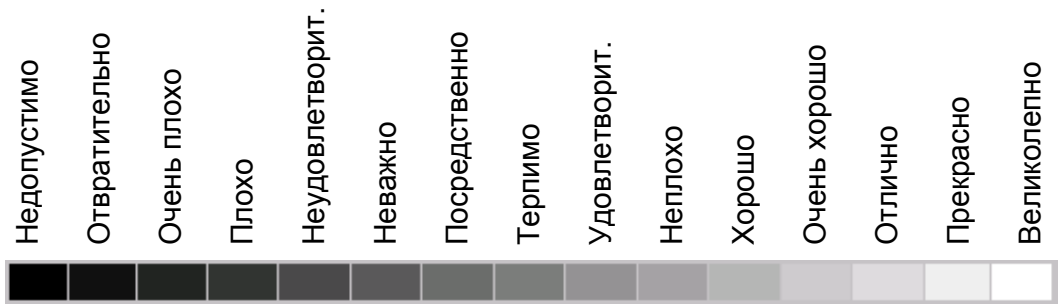


Рис. 2.3. Шкала предпочтений для свертки лексических показателей.

На рисунке 2.4. показано раскрашенное пользователем сечение при значении фиксированного показателя «Ресурсы МО», заданного лексической оценкой «отлично».

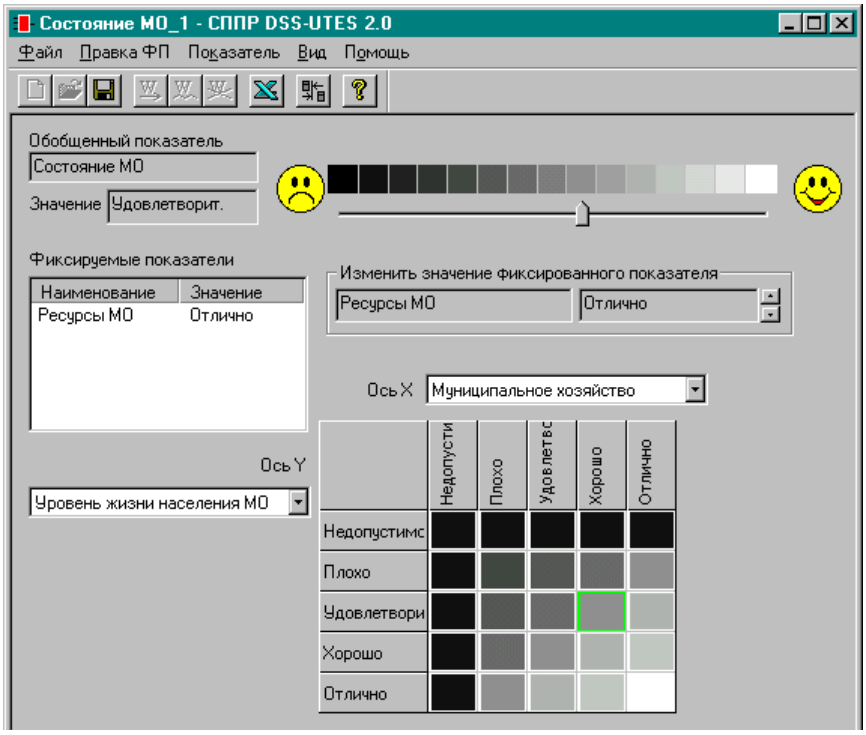


Рис. 2.4. Заполненное основное окно DSS/UTES («Ресурсы МО» – «отлично»).

Рисунок 2.5. повторяет предыдущий, но фиксированному показателю придано значение «плохо». Поэтому все прямоугольники стали «темнее». Так надо раскрасить все сечения для разных значений фиксированного показателя. В системе для облегчения работы пользователя эта процедура может быть частично автоматизирована. Окончание раскраски означает, что пользователь ввел скалярную функцию значений сворачиваемых показателей, вычисляющую обобщенный критерий. В дальнейшем, при изменениях значений компонент, значение обобщенного показателя будет вычисляться в автоматическом режиме. Выполнение этой процедуры по сути означает настройку АСМ МО на вычисление данного обобщенного показателя. Такие настройки надо произвести для всех обобщенных показателей, сворачиваемых на основе учета зависимости по предпочтениям.

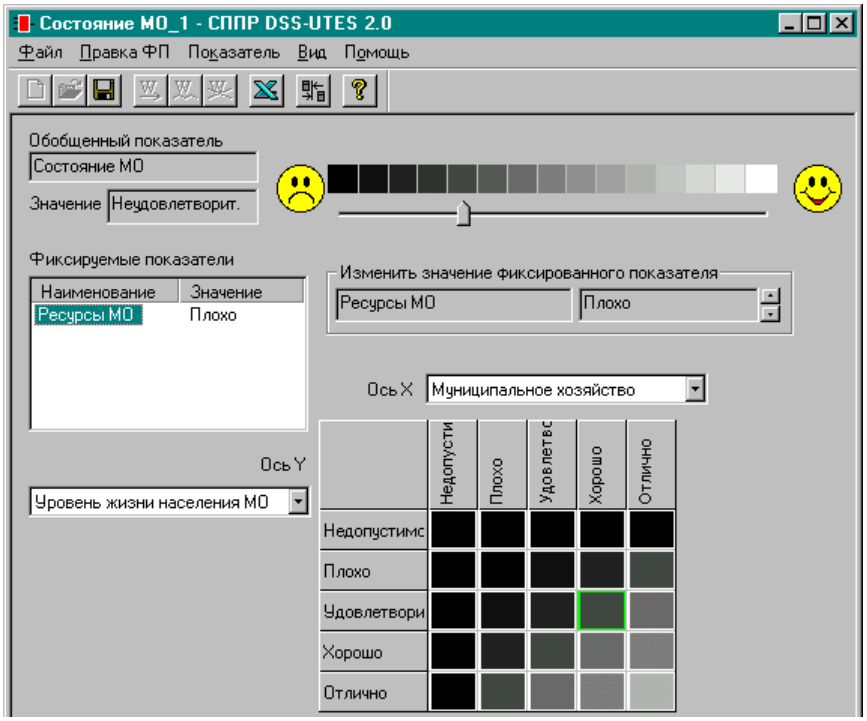


Рис. 2.5. Заполненное основное окно DSS/UTES («Ресурсы МО» – «плохо»).

Для каждой альтернативы может быть определено ее значение в шкале предпочтений. На рисунке 2.5. выделено значение обобщенного показателя для следующих значений агрегируемых:

- «Муниципальное хозяйство» – «хорошо»;
- «Уровень жизни населения» – «удовлетворительно»;
- «Ресурсы МО» – «плохо».

Вычисленное значение обобщенного показателя «Состояние МО» – «неудовлетворительно». Когда «Ресурсы МО» оценивались как «отличные», состояние МО было оценено как «удовлетворительное».

Теперь рассмотрим свертку обобщенного показателя на основе взвешенной суммы. Коэффициенты важности назначаются ЛПР или экспертами. Для приведения всех агрегируемых показателей к единой шкале допустимо их нормирование.

Наиболее принципиальной и трудной для пользователя операцией является назначение весовых коэффициентов. ЛПР может назначить их сам (что практически возможно только при невысокой мерности векторного критерия) или воспользоваться вспомогательной процедурой их вычисления, широко известной под названием «метод парных сравнений». Этот метод реализован в DSS/UTES 2.0. и, соответственно, в АСМ МО.

Метод парных сравнений имеет две разновидности: аддитивный и мультипликативный. Пусть a_{ij}, (i>j) - важность (вес) показателя i по отношению к показателю j. Вес показателя j относительно i можно определить двумя способами $a_{ji}=1- a_{ij}$, или $a_{ij}=1/ a_{i\ j}$. Первый способ

означает аддитивный метод парных сравнений, второй – мультипликативный. В содержательном отношении разница между ними состоит в том, что в аддитивном методе пользователь (эксперт) определяет **на сколько** один показатель важнее другого, а в мультипликативном - **во сколько .** Что касается формальной процедуры, то она одинакова для обеих методов, т.к. весовые коэффициенты вычисляются на основе собственного вектора мультипликативной матрицы, соответствующего ее максимальному собственному значению, а мультипликативная матрица связана с аддитивной однозначным, чисто формальным преобразованием. Поэтому выбор метода определяется исключительно тем, какой из них более удобен пользователю.

В диалоге АСМ МО с пользователем «по умолчанию» при назначении относительных весов показателей принят аддитивный метод.

Аддитивная матрица имеет вид:

$$A=\begin{bmatrix} 0{,}5 & a_{12} & \dots & a_{1n} \\ 1-a_{12} & 0{,}5 & \dots & a_{21} \\ \dots & \dots & \dots & \dots \\ 1-a_{1n} & 1-a_{2n} & \dots & 0{,}5 \end{bmatrix}$$

Пользователь на основе попарных сопоставлений назначает наддиагональные элементы, а симметричные (относительно диагонали) a_{ji} вычисляются как 1- a_{ij} . В случае i=j элементу присваивается значение 0,5.

Для проведения парных сравнений n показателей, количество определяемых пользователем отношений равно: (n*(n-1))/2. При высокой мерности критерия это много. Поэтому для ускорения ввода предпочтений предусмотрена процедура краткого опроса, при которой пользователь задает в каждом столбце матрицы А лишь по одному элементу, расположенному над диагональю, а именно, элементы a_{ij} , где j=i+1, i=1..n-1. Далее матрица парных сравнений приводится к мультипликативному виду. После чего, остальные элементы мультипликативной матрицы парных сравнений находятся из соотношения:

$a_{ij}=a_{i(i+1)}a_{(i+1)j}$, j=2..n, i=j-1..1.

В этом случае общее число парных сравнений равно n-1, матрица А полностью согласована (не содержит петель нетранзитивности), однако нельзя быть уверенным в том, что все вычисленные элементы отражают точку зрения пользователя. Последнее является платой за простоту, тем, что выбор способа заполнения матрицы предоставляется пользователю.

Методы получения оценок весов показателей на основе парных сравнений хорошо описаны в литературе [36, 48].

Процедура свертки векторного критерия на основе взвешенной суммы и парных сравнений, как и описанный выше режим учета зависимых предпочтений, выполняется в АСМ

МО после того, как определены все показатели, заданы направления их улучшения и введены минимальные и максимальные значения.

Для ветви СОП, приведенной на рисунке 2.1., свертка на основе взвешенной суммы может быть применена для вычисления обобщенного показателя «Качество незастроенных территорий», т.к. все его компоненты принадлежат одному типу (количественные) и измеряются в единой шкале.

На рисунке 2.6. показано окно метода парных сравнений в режиме краткого опроса, а на рисунке 2.7. - в режиме полного опроса. Оба режима выполнялись одним и тем же пользователем. Необходимо отметить, что краткий опрос занял около 15 минут, в то время как полный потребовал около 1,5 часов.

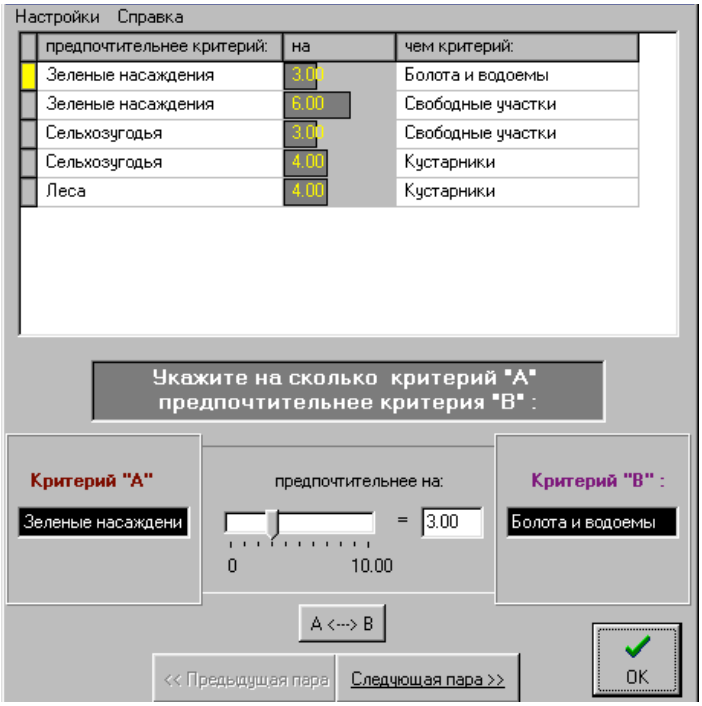


Рис. 2.6. Окно метода парных сравнений.

предпочтительнее критерий:	на	чем критерий:
Зеленые насаждения	3.00	Болота и водоемы
Болота и водоемы	0.00	Свободные участки
Сельхозугодья	3.00	Болота и водоемы
Болота и водоемы	2.00	Кустарники
Леса	4.00	Болота и водоемы
Зеленые насаждения	6.00	Свободные участки
Сельхозугодья	1.00	Зеленые насаждения
Зеленые насаждения	2.00	Кустарники
Зеленые насаждения	0.00	Леса
Сельхозугодья	3.00	Свободные участки
Свободные участки	4.00	Кустарники
Свободные участки	0.00	Леса
Сельхозугодья	4.00	Кустарники
Сельхозугодья	1.00	Леса
Леса	4.00	Кустарники

Рис. 2.7. Ввод парных сравнений в режиме полного опроса.

В таблице 2.1. приведена мультипликативная матрица парных сравнений, отражающая результат краткого опроса, а в таблице 2.2. - результат полного опроса.

Табл. 2.1.

	болота и водоемы	зеленые насаждения	свободные участки	сельхоз-угодия	кустарники	леса
болота и водоемы	1,00	0,54	2,15	1,16	2,71	1,16
зеленые насаждения	1,86	1,00	4,00	2,15	5,03	2,15
свободные участки	0,46	0,25	1,00	0,54	1,26	0,54
сельхозугодия	0,86	0,46	1,86	1,00	2,33	1,00
кустарники	0,37	0,20	0,80	0,43	1,00	0,43
леса	0,86	0,46	1,86	1,00	2,33	1,00

Табл. 2.2.

	болота и водоемы	зеленые насаждения	свободные участки	сельхоз-угодия	кустарники	леса
болота и водоемы	1,00	0,54	1,00	0,54	1,50	0,43
зеленые насаждения	1,86	1,00	4,00	0,82	1,50	1,00
свободные участки	1,00	0,25	1,00	0,54	2,33	1,00
сельхозугодия	1,86	1,22	1,86	1,00	2,33	1,22
кустарники	0,67	0,67	0,43	0,43	1,00	0,43
леса	2,33	1,00	1,00	0,82	2,33	1,00

Из сопоставления матриц видно, что веса, назначаемые пользователем непосредственно, отличаются от вычисляемых в процедуре краткого опроса. Однако это различие может

оказаться несущественным, если вычисленные на их основе коэффициенты важности будут достаточно близки в том смысле, что полученные в конечном счете оценки обобщенных показателей будут различаться в пределах градаций соответствующих шкал.

Заметим, что матрица парных сравнений, вычисленная в процедуре краткого опроса, является согласованной по определению и в ней исключено наличие нетранзитивных оценок. В то же время матрица, полученная путем непосредственных сравнений всех показателей, характеризуется коэффициентом согласования 0,6, что говорит о наличии в ней несогласованных суждений [70]. Естественно, что делать сколько-нибудь общие выводы на основании единичного примера нельзя, однако сам факт достаточно примечателен и указанное обстоятельство необходимо учитывать при настройке процедур вычисления обобщенных показателей АСМ МО.

В качестве метода получения полностью согласованной матрицы парных сравнений в работе предлагается следующий итеративный способ согласования методов назначения по строке и диагонали. В нашем примере пользователь задал только одну первую строку матрицы парных сравнений и по ней система восстанавливает наддиагональные элементы. Таким образом, формируется первый вариант матрицы $A^{1}3$ (верхний индекс показывает номер итерации), показанный в таблице 2.3.

Табл. 2.3.

	болота и водоемы	зеленые насаждения	свободные участки	сельхоз-угодия	кустарники	леса
болота и водоемы	1,00	0,54	1,00	0,54	1,50	0,43
зеленые насаждения		1,00	1,86			
свободные участки			1,00	0,54		
сельхозугодия				1,00	2,79	
кустарники					1,00	0,29
леса						1,00

Далее пользователь корректирует назначения, сделанные над главной диагональю, и система вычисляет новые значения весов в первой строке, которые снова предъявляются пользователю для корректировки и т.д. Признаком окончания выполнения данной процедуры служит согласие пользователя с пересчетом системы из диагонали в первую строку или наоборот. Практика показала, что данная процедура сходится за 4-6 итераций.

В результате применения данной процедуры на четвертом шаге получили следующую матрицу парных сравнений, отвечающую представлениям об относительной важности показателей - таблица 2.4.

Табл. 2.4.

	болота и водоемы	зеленые насаждения	свободные участки	сельхоз-угодия	кустарники	леса
болота и водоемы	1,00	0,53	1,59	0,54	1,35	0,70
зеленые насаждения	1,89	1,00	3,00	1,02	2,55	1,32
свободные участки	0,63	0,33	1,00	0,34	0,85	0,44
сельхозугодия	1,85	0,98	2,94	1,00	2,50	1,29
кустарники	0,74	0,39	1,18	0,40	1,00	0,52
леса	1,43	0,76	2,27	0,77	1,93	1,00

Данная матрица парных сравнений является полностью согласованной.

В таблице 2.5. даны значения исходных показателей для тестового расчета. В таблице 2.6. приведены вычисленные по методу взвешенной суммы значения обобщенного показателя «Качество незастроенных территорий» для трех вышеописанных режимов формирования матрицы парных сравнений.

Табл. 2.5. Исходные данные.

Болота и водоемы (га.)	10
Зеленые насаждения (га.)	2
Свободные участки (га.)	5
Сельхозугодия (га.)	13
Кустарники (га.)	13
Леса (га.)	17

Табл. 2.6. Весовые коэффициенты и результат мониторинга.

	Режим краткого опроса	Режим полного опроса	Эвристический режим опроса
Болота и водоемы	0,1847	0,1182	0,1327
Зеленые насаждения	0,3429	0,2261	0,2504
Свободные участки	0,0857	0,1289	0,0835
Сельхозугодия	0,1592	0,2367	0,2455
Кустарники	0,0682	0,0894	0,0982
Леса	0,1592	0,2006	0,1896
Результаты мониторинга			
Незастроенные территории	0,6767	0,6028	0,6068
Качество незастроенных территорий	хор.	хор.	хор.

Далее, в соответствии с СОП, нужно перейти к лексическому показателю «Качество незастроенных территорий». Способы построения шкал рассмотрены в главе 3. Если принять самый простой - линейное разбиение, то это будет соответствовать введению бальной оценки с одинаковыми градациями. Например, шаг 0,2 пятибалльная шкала, или 0,1 десятибалльная

шкала. Пятибалльная шкала может измеряться в следующих термах: «недопустимо», «плохо», «удовлетворительно», «хорошо», «отлично». Из таблицы видно, что изменение режима опроса приводит к одному и тому же результату вплоть до десятибалльной шкалы. Применение шкал с большем количеством градаций как правило не употребительно. Таким образом, различие между методами опроса в рассматриваемой задаче малосущественно, в пределах одной градации десятибалльной шкалы, и пользователь может выбирать наиболее удобный для себя метод. Однако этот вывод делается на основании единичного примера и, естественно, недостаточно убедителен. Поэтому в качестве рекомендуемого разумно принять режим итерационного опроса, сочетающий в себе краткость и транзитивность. Кроме того, необходимо иметь в виду, что различие в оценках при многократном огрублении, как правило, не сказывается на верхнем уровне дерева показателей. Это обстоятельство отражает характерное для иерархических структур распределение функций управления по уровням иерархии.

Рассмотренный пример имеет еще один аспект. Из содержательного анализа оценки качества незастроенных территорий видно, что одно и то же числовое значение может быть получено при различных сочетаниях исходных данных, и поэтому балльная оценка, вычисленная как среднее взвешенное, не отражает реального состояния объекта. Очевидно, что если лесов много, а свободных территорий мало, то можно поступиться лесом во имя получения площадей под застройку. И наоборот, если леса и зеленых насаждений мало, то часть свободных территорий надо отвести под озеленение. Здесь очень хорошо видно, что важности (веса) показателей качества незастроенных территорий взаимозависимы, и эта зависимость показателей является зависимостью по предпочтениям. Отсюда следует вывод, что несмотря на то, что показатели качества незастроенных территорий носят числовой характер, их свертку надо производить не методом взвешенной суммы, а использовать основной режим DSS/UTES – свертка с учетом зависимости по предпочтениям. Таким образом, приведенный пример лишний раз подтверждает, что выбор способа свертки обобщенных критериев требует тщательного содержательного анализа.

2.1.2. Работа пользователя с интерфейсом АСМ МО

В этом параграфе приводится краткое описание основных элементов интерфейса АСМ МО с тем, чтобы создать общее представление о функционировании системы и правилах работы с ней. В АСМ МО по умолчанию прописана структура основных показателей (СОП), разработанная в Академии им. Плеханова. Поэтому на первом этапе работы с АСМ МО необходимо настроить ее на конкретное муниципальное образование (МО), т.е.

отредактировать перечень и структуру показателей. Соответствующее окно представлено на рисунке 2.8.

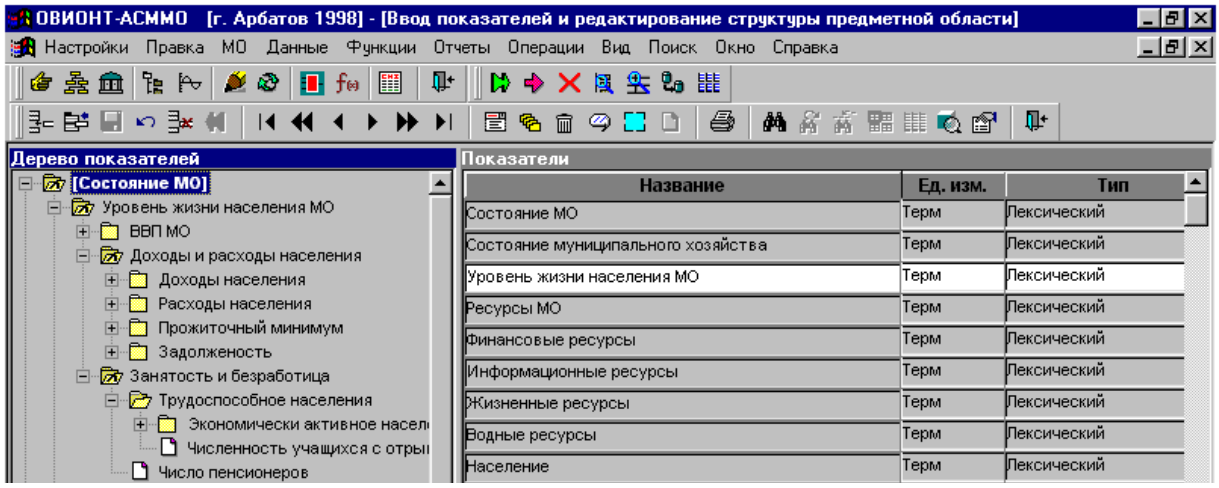


Рис. 2.8. Окно редактирования дерева показателей.

Каждый показатель имеет такие атрибуты как тип, единицы измерения, шкалу, источник. Список возможных значений этих атрибутов редактируется в справочниках. Пример окна справочника типов приведен на рисунке 2.9.

Название источника	Описание источника
Расчетный	
СтУ	Статистическое управление
КЗР	Комитет по земельным ресурсам
КЭ	Комитет по экономике
ФУ	Финансовое управлениие
БТИ	Бюро технической инвентаризации

Рис. 2.9. Редактирование источников.

Ввод значений показателей может осуществляться тремя способами: ввод значений показателей в специальном окне АСМ МО; ввод значений показателей, заданных в текстовом файле; получение значений показателей по электронной почте. Последние две возможности реализованы в рамках специальной подсистемы экспорта-импорта информации. Данная подсистема позволяет организовать пересылку данных с одного уровня иерархии управления на другой. Т.е. можно передавать информацию о значениях показателей из БД МО на областной и федеральный уровень. В системе предусмотрены возможности по контролю вводимых данных. Для этого задаются логические условия проверки корректности значений показателей.

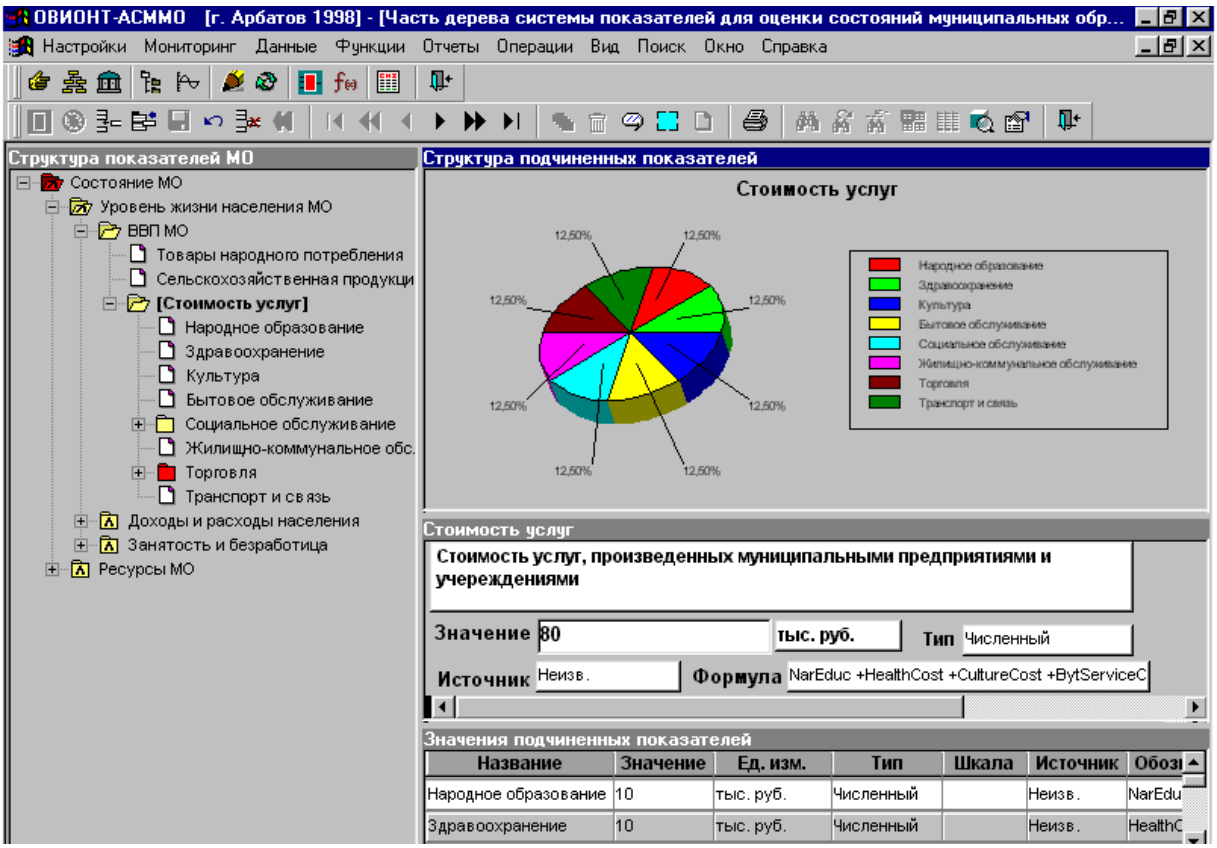


Рис. 2.10. Мониторинг МО.

Пользователь может наблюдать текущее состояние МО в окне мониторинга (см. рисунок 2.10.). В этом окне отображается структура показателей МО, структура подчиненных показателей, значение выбранного показателя, значения подчиненных показателей. АСМ МО позволяет проводить графическое сопоставление значений показателя в различные моменты времени. Как непрерывно, так и по временным срезам. Для этого используется окно сравнения состояний, изображенное на рисунке 2.11. В этом окне отображается структура показателей и изменение во времени значений выбранного показателя и подчиненных ему показателей.

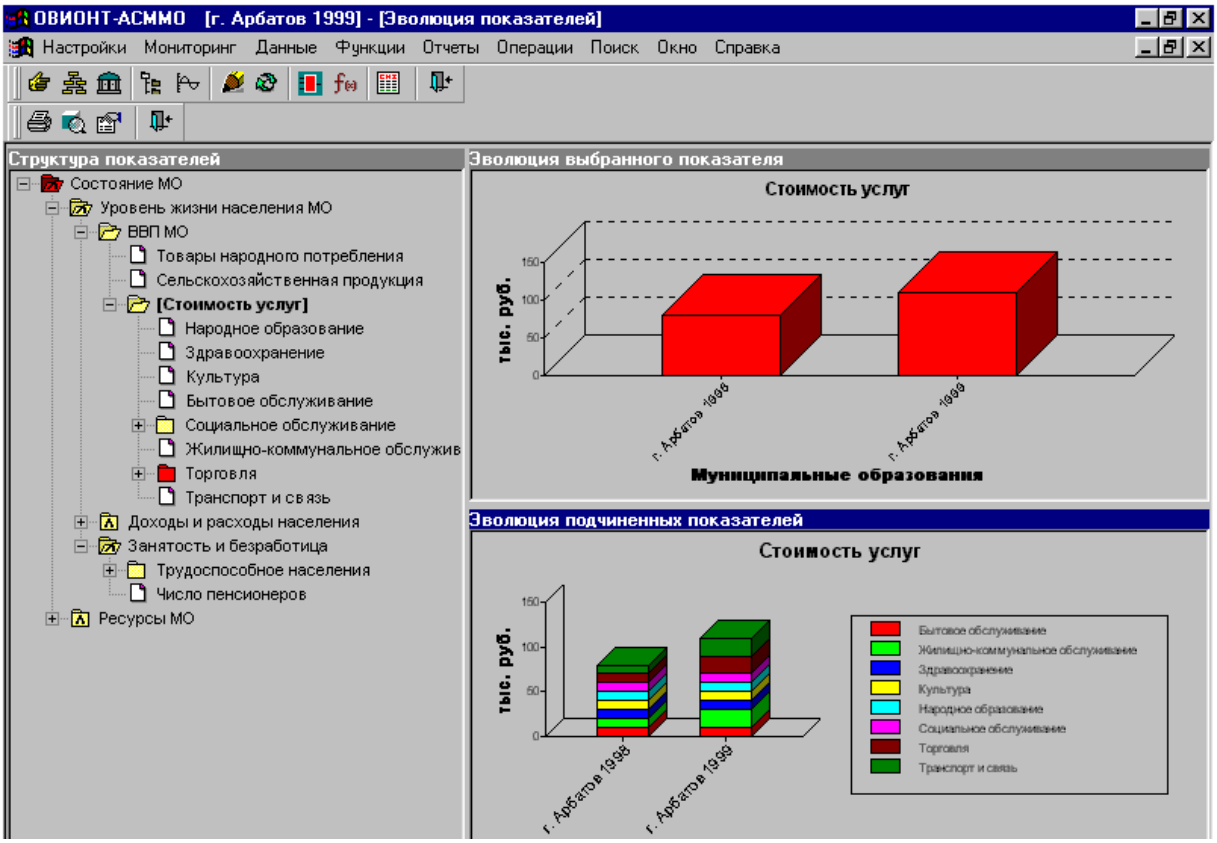


Рис. 2.11. Динамика показателей.

Для неформального анализа результатов мониторинга созданы экспортируемые в MS Excel отчеты: ОФ-001 - отчет об иерархии показателей; ОФ-002 - отчет о значениях показателей; ОФ-003 - отчет о значениях показателей в заданном временном диапазоне; ОФ-004 - отчет о сравнительных значениях выбранных показателей для выбранных МО.

2.1.3. Управление на основе мониторинга

Рассмотрим кратко возможности управления МО на основе мониторинга. При построении шкал обобщенных показателей использовалась процедура «ручного загрубления». Пользователь сам выбирал, какие градации «загрубленной» шкалы соответствуют градациям исходной шкалы ФП. Данные для загрубления по тестовой ветке приведены на рисунке 2.12. Последний показатель «Состояние МО» имеет невысокую степень загрубления, т.к. в дальнейших операция свертки он не участвует и пользователю желательно максимально сохранить содержащуюся в нем информацию.

Имя показателя	Мерность незагруб-ленной шкалы	Мерность загрубленной шкалы	Градации «загрубленной» шкалы
Состояние МО	150	16	недопустимо, отвратительно, очень плохо, плохо, неудовлетворительно, неважно, посредственно, терпимо, удовлетворит., неплохо, хорошо, очень хорошо, отлично, прекрасно, великолепно
Муниципальное хозяйство	200	5	недопустимо, плохо, удовлетворительно, хорошо, отлично
Уровень жизни населения	50	5	недопустимо, плохо, удовлетворительно, хорошо, отлично
Ресурсы МО	20	6	недопустимо, плохо, удовлетворительно, терпимо, хорошо, отлично
Финансовые ресурсы	20	2	недостаточно, достаточно
Информационные ресурсы	6	2	недостаточно, достаточно
Жизненные ресурсы	30	5	недопустимо, плохо, удовлетворительно, хорошо, отлично
Водные ресурсы	10	2	недостаточно, достаточно
Население (демографическое состояние)	12	3	недопустимо, терпимо, хорошо
Земельные ресурсы	25	5	недопустимо, плохо, удовлетворительно, хорошо, отлично
Незастроенные территории (качество)	5	5	недопустимо, плохо, удовлетворительно, хорошо, отлично
Болота и водоемы	от 0 до 20 га. (непрерывная)		
Зеленые насаждения	от 0 до 20 га. (непрерывная)		
Свободные участки	от 0 до 20 га. (непрерывная)		
Сельхозугодия	от 0 до 20 га. (непрерывная)		
Кустарники	от 0 до 20 га. (непрерывная)		
Леса	от 0 до 20 га. (непрерывная)		
Застроенные территории (качество)	5	5	недопустимо, плохо, удовлетворительно, хорошо, отлично

Рис. 2.12. Шкалы обобщенных показателей.

На рисунке 2.13. показано изменение состояния по двум временным срезам ветви «Состояние МО - Ресурсы МО - Жизненные ресурсы - Земельные ресурсы - Незастроенные территории – (*Болота и водоемы (га.), Зеленые насаждения (га.), Свободные участки (га.), Сельхозугодия (га.), Кустарники (га.) Леса (га.)*)».

В этом примере временные срезы разделены двухгодовым интервалом. Изменения обусловлены тем, что за прошедшие два года было вырублено 9 га. лесов, осушена половина

болот и освоено 17 га. под новые зеленые насаждения, а часть кустарников и сельхозугодий освободилась под свободные участки и зеленые насаждения. Это изменило оценку качества незастроенных территорий и, в конечном счете, повысило оценку состояния МО с «терпимо» до «удовлетворительно».

Имя показателя	12.12.1997	12.12.1999	
Состояние МО	Терпимо	Удовлетворит.	
Муниципальное хозяйство	Хорошо	Хорошо	
Уровень жизни населения	Хорошо	Хорошо	
Ресурсы МО	Терпимо	Хорошо	
Финансовые ресурсы	Недостаточно	Недостаточно	
Информационные ресурсы	Достаточно	Достаточно	
Жизненные ресурсы	Удовлетворит.	Хорошо	
Водные ресурсы	Достаточно	Достаточно	
Население (демографическое состояние)	Хорошо	Хорошо	
Земельные ресурсы	Удовлетворит.	Хорошо	
Незастроенные территории (качество)	0,4969 (Удовлетворит.)	0,6068 (Хорошо)	
Болота и водоемы (га.)	10	5	
Зеленые насаждения (га.)	2	19	
Свободные участки (га.)	5	14	
Сельхозугодия (га.)	13	9	
Кустарники (га.)	13	5	
Леса (га.)	17	8	
Застроенные территории (качество)	0,51 (Хорошо)	0,51 (Хорошо)	

* - условное значение обобщенного показателя (заданное в примере)

Рис. 2.13. Изменение обобщенных показателей.

Произошедшие изменения можно рассматривать как результат реализации управленческого решения. Таким образом, АСМ МО позволяет моделировать последствия принимаемых решений не выходя из кабинета и затрачивая минимум времени. Для этого достаточно вводить в систему новые значения показателей, которые предполагается достичь в результате управления. Принципиальным моментом такого моделирования является заложенный в нем системный подход, в том смысле, что все показатели взаимосвязаны и изменение любого из них автоматически отразится на всех других, на которые он явно или неявно (с точностью до структуры) влияет. На практике подобное влияние далеко не всегда можно не только рассчитать, но даже предусмотреть.

Рассматривая приведенный пример, следует иметь в виду, что он носит чисто иллюстративный характер, т.к. предпочтения вводились не квалифицированными экспертами, а

автором работы. При этом неявно подразумевалось, что МО имеет характер территориального образования примыкающего к крупному городу.

2.1.4. Программная реализация и внедрение АСМ МО

При выборе системы программирования клиентской части одним из важных факторов было наличие средств написания многофункционального интерфейса пользователя и наличие готовых инструментов формирования запросов к СУБД MS SQL Server [136]. Использование при создании СППР DSS/UTES технологии COM позволяет интегрировать DSS/UTES с большинством современных систем программирования. Выбор системы программирования АСМ МО целесообразно было осуществлять среди хорошо известных и распространенных продуктов. С малоизвестными системами программирования велик риск в будущем лишиться возможности получать новые версии системы программирования под новые операционные системы.

В наибольшей степени выдвинутым требованиям удовлетворяет PowerBuilder компании Sybase [66]. В PowerBuilder встроены свои компоненты доступа к БД. Эти компоненты не требуют предварительной установки на клиентском компьютере, что является преимуществом перед другими известными компонентами доступа к БД, такими как BDE и ADO [26]. В результате, программное обеспечение АСМ МО написано в среде PowerBuilder. Стартовой точкой программы является модуль asmmo.exe. Он выполняет запуск виртуальной машины PowerBuilder, на которой исполняются основные модули системы (*.pbd). Основные модули системы представляют собой псевдокод виртуальной машины. Таким образом, проблемная часть программы становиться независимой от ОС. Меняя модуль виртуальной машины можно переходить на другие ОС.

На рисунке 2.14. показана обобщенная диаграмма развертывания АСМ МО. На верхнем уровне иерархии находится ряд рабочих мест соединенных локальной сетью. Вся информация храниться в БД под управлением MS SQL Server [136]. Исходные данные поступают от подчиненных МО в виде текстовых файлов. Эти файлы могут приниматься почтовой службой и поступать в БД автоматически. Возможно также получить непосредственный доступ к БД с удаленного компьютера через Internet. Администратор системы контролирует процесс информационного наполнения системы и отвечает за резервное копирование данных. Начальники департаментов и руководители на этапе ввода системы в эксплуатацию согласовывают дерево показателей и вводят предпочтения в узлах свертки, а в процессе эксплуатации системы получают различные срезы по обобщенным показателям.

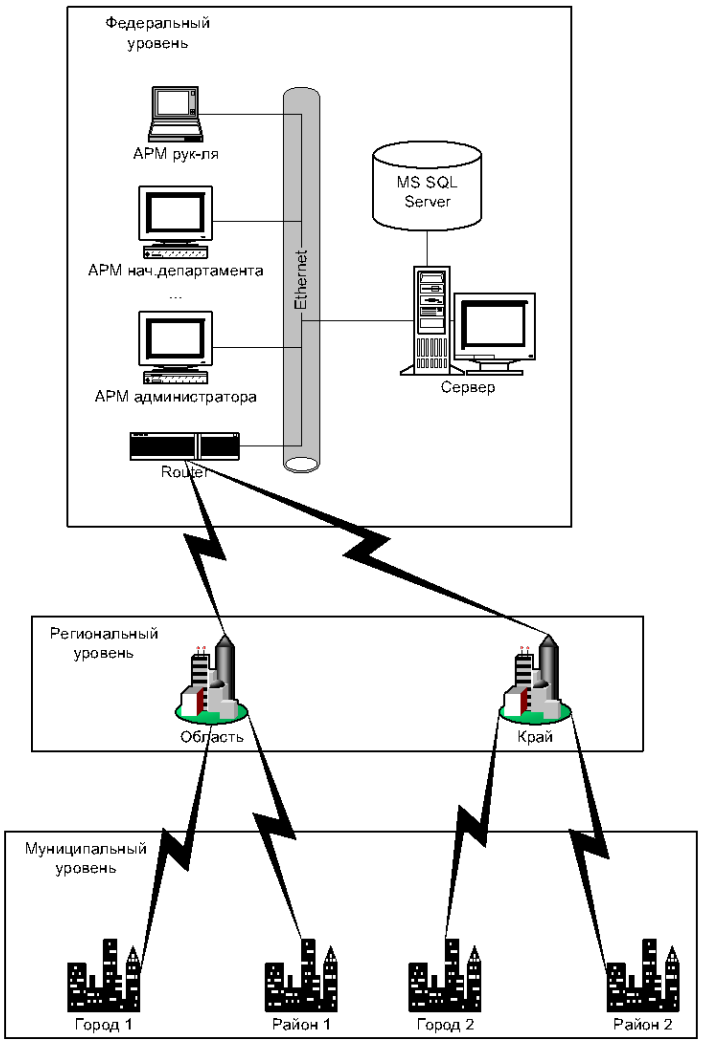


Рис. 2.14. Диаграмма развертывания АСМ МО.

Аналогичная сетевая структура может быть создана на региональном и муниципальном уровне. Ввод данных в системы может осуществляться как из электронных источников, так и непосредственно в окнах программы. Если территория не располагает минимальными аппаратными и программными ресурсами, то для ввода данных на верхний уровень иерархии управления могут передаваться файлы, созданные вручную. Обязательным требованием к внедрению является наличие на территории квалифицированных администраторов.

АСМ МО установлена во Всероссийском центре муниципальной информации в Министерстве по делам федерации, национальной и миграционной политике, в администрациях городов Печоры, Иркутск, Иваново, Ржев, Дзержинский.

2.2. «Автоматизированная система контроля и управления (АСКУ)»

АСКУ предназначена для мониторинга состояния иерархических систем по многим показателям, имеющим древовидную структуру [64, 129]. Древовидный характер структуры показателей, с одной стороны, определяется иерархической структурой системы, а, с другой стороны, древовидным характером показателей для каждого объекта системы. Объекты на

одном уровне иерархии могут быть разнотипными, т.е. дерево показателей для них может быть своё. Дерево показателей для каждого объекта иерархической структуры определяется пользователем АСКУ. Отдельные показатели могут быть как числовыми, так и лексическими. Корень всей древовидной структуры показателей соответствует обобщенной оценки состояния всей оцениваемой системы, а листва представляет так называемые исходные показатели. Они не являются обобщенными, вычисляемыми, а задаются пользователем, как исходные данные для оценки состояния всей системы и отдельных её частей (объектов). Пользователи помимо древовидной структуры показателей определяют правило вычисления обобщенных показателей на основе значений подчиненных показателей. Для определения этих правил в АСКУ могут использоваться возможности системы DSS/UTES, либо формулы, имеющие логический характер. Оценка состояния системы заключается в расчете всех обобщенных показателей. В общем случае эта оценка может проводиться для различных моментов времени (временных срезов). При сопоставительном анализе оценок для различных временных срезов АСКУ позволяет выявлять причины изменения обобщенных показателей отдельных объектов и системы в целом. АСКУ позволяет проводить ранжировку однотипных по составу показателей объектов иерархической структуры по заданному показателю. Для осуществления всех перечисленных операций АСКУ обладает следующим набором функций:

- задание и хранение иерархической структуры оцениваемой системы;
- формирование и хранение древовидной структуры показателей отдельных объектов;
- построение и анализ полного дерева показателей для всей оцениваемой системы;
- задание и хранение в базе данных значений исходных показателей, определяемых либо в числовой, либо в лексической шкалах (в том числе, и по временным срезам);
- задание правил вычисления обобщенных показателей;
- вычисление, хранение и представление пользователю в удобной форме для анализа значений обобщенных показателей (в том числе, и по временным срезам);
- проведение анализа причин изменения обобщенных показателей путем обратного движения по дереву показателей от обобщенных к исходным;
- ранжировка объектов по заданному обобщенному показателю.

Система, состояние которой оценивается, состоит из множества объектов, объединенных в иерархическую структуру, определяющую отношение подчиненности. Каждый из объектов характеризуется совокупностью показателей, по которым производится оценка его состояния. Сами показатели образуют древовидную структуру, в которой значения показателей верхних уровней вычисляются на основе значений показателей нижних уровней. Программа позволяет вычислять значения всех обобщенных показателей. Для этого первоначально вводятся исходные показатели (первичные, не вычисляемые), они объединяются в группы и задается

правило вычисления соответствующих обобщенных показателей. Обобщенные показатели, в свою очередь, могут также объединяться (агрегироваться) в группы, которые выступают в качестве новых обобщенных показателей и т.д. Таким образом, создается дерево показателей, которое позволяет получить общую оценку объекта и системы в целом.

Перед вводом в эксплуатацию программа требует настройки. Для этого вводятся данные определяющие:

- типы возможных объектов оценки;
- дерево показателей для соответствующих типов объектов;
- способы и параметры свертки показателей для соответствующих типов объектов;
- иерархию объектов.

В процессе оценки состояния объектов в качестве входных данных выступают значения исходных показателей на определенный момент времени. В качестве выходных данных выступают:

- значения обобщенных показателей;
- графическое представление значений обобщенных показателей;
- табличное представление результатов оценки для нескольких временных срезов;
- ранжировка объектов по выбранному обобщенному показателю.

2.2.1. Правила построения обобщенных оценок на основе формальных правил

Ряд показателей многоуровневого дерева агрегирования вычислялся в АСКУ на основе формальных правил и должностных инструкций. Говорить о системе ценностей пользователя в СППР, работающей по подобным инструкциям, не приходится. Хотя эти инструкции и представляют некую систему ценностей их составителей.

В принципе можно было бы использовать традиционный интерфейс ФП DSS/UTES для записи формальных правил, т.к. любая объективная ФП задает упорядочивание полного множества альтернатив. Однако это нецелесообразно, т.к. интерфейс плоских сечений неудобен оператору для ввода формальных правил, и поэтому высока вероятность допустить ошибку при ручном переводе правила в картинку. Отсюда возникает задача построения автономной подсистемы для задания объективных ФП. С этой целью можно воспользоваться одним из следующих подходов:

1. Реализовать необходимые правила в рамках основной программы.
2. Реализовать возможность динамического подключения к программному обеспечению дополнительных модулей, отвечающих за необходимые правила.
3. Реализовать подключение языка интерпретирующего типа и задавать правила вычисления на этом языке в виде небольших модулей.

4. Реализовать средства транслирования правил вычислений в рамках собственной грамматики.

Главный недостаток первого подхода очевиден: для изменения правил вычислений приходится перестраивать основную программу. Поэтому все правила должны быть определены на момент написания основной программы. Правила вычислений должны меняться редко и служба поддержки ПО должна быть способна самостоятельно изменять основную программу.

Второй подход позволяет задавать спецификацию подключения, например, такую как в DSS/UTES 2.0 (см. первую главу настоящей работы и [149]). Сейчас такая спецификация создается в рамках какого-либо стандарта, например, COM или COBRA. Сложность решающих правил в данном случае ограничена только возможностями интерфейса программного доступа и языком программирования, используемым для написания подключаемых компонентов. Недостатком данного похода является то, что для написания решающих правил приходится привлекать программистов, имеющих высокую квалификацию.

Третий подход широко распространен в рамках офисных приложений фирмы Microsoft. В качестве языка интерпретатора используется Visual Basic for Applications. Такой подход позволяет реализовать достаточно сложные алгоритмы вычислений и не требует от программиста высокой квалификации. Недостатком данного подхода является сложность организации вызова интерпретатора из основной программы.

Четвертый подход позволяет в большинстве случаев отказаться от программиста при задании правил вычислений. Суть его заключается в применении языка арифметических и логических формул, который понятен квалифицированному пользователю, не обязательно программисту. Для проведения условных вычислений предлагается использовать широко известные в ЭС правила продукций [32].

Перед разработкой формальной грамматики были проанализированы различные варианты формальных правил вычислений оценок, что позволило реализовать грамматику наиболее компактно и просто.

Для описания языка ввода объективных ФП воспользуемся метаязыком Бэкуса [54]. Данный метаязык отличается компактностью, простотой записи и легкостью дальнейшего программирования. Основные правила языка имеют следующий вид:

*p1: <буква>→q|w|e|r|t|y|u|i|o|p|a|s|d|f|g|h|j|k|l|z|x|c|v|b|n|m|й|ц|у|к|е|н|г|ш|щ|з|х|ъ|ф|ы|в|а|п|р|о|л|д|ж|э|я|ч|с|м|и|т|ь|б|ю|_*

p2: <цифра>→1|2|3|4|5|6|7|8|9|0

p3: <операция>→+|-||/|=|<|>|>=|<=*

p4: <знак>→+|-

p5: <идентификатор>→<буква>

p6: <идентификатор>→<идентификатор><буква>

p7: <идентификатор>→<идентификатор><цифра>

p8: <константа>→<цифра>

p9: <константа>→<знак><цифра>

p10: <константа>→<константа><цифра>

p11: <константа>→<константа>.<цифра>

p12: <имя объекта>→[<идентификатор>]\[]

p13: <имя показателя>→[<идентификатор>]\[]

p14: <имя показателя>→<имя объекта>.<имя показателя>

p15: <имя показателя>→.<имя показателя>

p16: <имя функции>→max|min|sum|case|count|if|and|or|not

p17: <функция>→<имя функции>(<список аргументов>)

p18: <список аргументов>→<пусто>

p19: <список аргументов>→<выражение>

p20: <список аргументов>→<список аргументов>;<выражение>

p21: <выражение>→

<имя показателя>|<функция>|<константа>|<лек.константа>

p22: <выражение>→<выражение><операция><выражение>

p23: <выражение>→(<выражение>)

p24: <лек.константа>→”<строка>”

p25: <строка>→<буква>|<цифра>|<знак>

p26: <строка>→<строка><строка>

Правила p1 и p2 определяют множество допустимых символов, используемых для записи идентификаторов показателей и оцениваемых объектов.

Правило p3 задает множество допустимых арифметических и логических операций.

Правило p4 определяет знаки, которые могут предшествовать числовой константе.

Правила p5, p6 и p7 означают: идентификатор - это строка букв и цифр, начинающаяся с буквы.

Правила p8-p11 говорят, что константа - это последовательность цифр, которая может начинаться со знака, и допустимо ставить в константе знак десятичной точки для отделения дробной части.

Правило p12 определяет, что оцениваемый объект записывается как идентификатор, если в квадратных скобках не указан идентификатор, то имеется ввиду текущий объект.

Правило p13 говорит, что имя показателя - это идентификатор, если используется показатель для того же объекта, что и вычисляемый, если в скобках пропущен идентификатор, то имеются в виду все дочерние показатели.

Правило p14 дает возможность применять составные имена для ссылки на показатели других объектов. Для ссылки на все подчиненные объекты можно использовать правило p15.

Список возможных имен функций задает правило p16. Определения функций будут даны ниже. Для устранения неоднозначности в качестве идентификаторов запрещается использовать имена функций.

Правило p17 определяет общую запись функции с обязательным указанием скобок. В скобках может быть список аргументов, разделенных точкой с запятой, но он может быть и пустым (см. правила p18-p20). Допустимы функции с любым количеством аргументов. В качестве аргументов функции выступают выражения (см. правила p19-p20).

Выражение представляет собой либо имя показателя, либо функцию, либо константу (см. правило p21). В качестве выражения может выступать комбинация двух выражений разделенная операцией (см. правило p22). Для указания порядка вычисления выражений любое выражение может быть заключено в скобки (см. правило p23).

Правила p24-26 определяют правила записи лексических констант. В кавычках может указываться любой символ кроме кавычки.

Рассмотрим основные функции определенного выше языка вычислений показателей, необходимые для реализации формальных правил, заложенных в инструкции:

«**or()**» - функция, реализующая операцию логического **ИЛИ** двух аргументов, являющихся логическими выражениями;

«**and()**» - функция, реализующая операцию логического **И** двух аргументов, являющихся логическими выражениями;

«**not()**» - функция, реализующая операцию логического отрицания одного аргумента, являющегося логическим выражением;

«**if()**» - конструкция ветвления, которая при выполнении логического выражения возвращает одно значение, в противном случае другое;

«**case()**» - конструкция ветвления, которая в зависимости от значения, которое принимает определенное выражение, может вернуть одно из нескольких выражений;

«**sum**» - сумма численных показателей, подчиненных данному показателю;

«**sum([_имя_])**» - сумма численных показателей с именем _имя_, относящихся к объектам, подчиненным данному объекту;

«**count(=|<|>|<=|>=|<> выражение)**» - счетчик показателей, являющихся дочерними, значения которых находятся в указанном отношении со значением выражения;

«**count([_имя_] =|<|>|<=|>=|<> выражение)**» - счетчик показателей с именем _имя_, относящихся к объектам, подчиненным данному объекту, значения которых находятся в указанном отношении со значением выражения;

«**max()**» – возвращает максимальное значение из всех значений своих аргументов;

«**min()**» – возвращает минимальное значение из всех значений своих аргументов. Число аргументов может быть любым.

Предполагается, что регистр при записи имен функций значения не имеет. Арифметические операции (+, -, *, /) допускаются только с числовыми значениями (в данном случае сумма двух операторов case допускается, т.к. оба возвращают только числовые значения). Операции сравнения (>, <, =, >=, <=, <>) допускаются только если оба операнда одинакового типа (оба численные или лексические). Если операнды - лексические, то результат сравнения является результатом сравнения рангов констант в шкале.

Рассмотрим пример разбора формулы:

case([организация подготовки безопасности]; ("Удовлетворительно"); 1; ("Хорошо"); 2; ("Отлично"); 3; 0)

В результате если значение показателя «организация подготовки безопасности» равно "Удовлетворительно", то case возвращает 1, если равно "Хорошо", то case возвращает 2, если равно "Отлично", то case возвращает 3, если значение показателя не равно ни одной из этих констант, то case возвращает 0.

2.2.2. Программное обеспечение АСКУ

При выборе системы программирования соображения были такими же как и при выборе системы программирования для реализации АСМ МО. Основное ПО написано на языке Power Script в среде быстрой разработке приложений Power Builder 6.5 фирмы Sybase [66]. Для увеличения быстродействия системы часть кода перенесена на сервер: ПО содержит хранимые процедуры, исполняемые на сервере под управлением Oracle 8.0 и написанные на PL SQL. Недостатком такой структуры является то, что если возникнет необходимость переноса программы на другой сервер баз данных, то хранимые процедуры придется переписывать заново, поэтому при практической реализации систем мониторинга надо искать компромисс между увеличением быстродействия и риском необходимости перепрограммирования реализованных на сервере процедур [64].

2.2.3. Работа пользователя с интерфейсом АСКУ

Наиболее принципиальным моментом интерфейса АСКУ отличающим его от интерфейса АСМ МО является редактор формальных правил. На рисунке 2.15. показано окно редактирования. Оно предназначено для составления формул логического и арифметического пересчета правил в функцию предпочтений.

Работа с окном заключается в следующем. При вызове редактора формул в него загружается формула, соответствующая выбранному обобщенному показателю, а в заголовке указывается его название.

В верхней панели окна редактора формул в многостроковом режиме отображается текст формулы. Его можно редактировать напрямую. Слева отображается структура показателей для текущего шаблона. Выбранный в этой структуре показатель можно вставить в формулу клавишей «**INSERT**» или перенести при помощи мыши в формулу. Справа имеется две панели со шкалами и градациями. Градации выводятся для выбранной шкалы. Она служит для облегчения построения символьных выражений. Любое значение можно перенести в формулу при помощи мыши и вставить выбранный элемент при помощи клавиши «**INSERT**». При выборе показателя в дереве для него автоматически отображается выбранная шкала. Также имеется набор кнопок с шаблонами лексем для облегчения построения формул при помощи мыши.

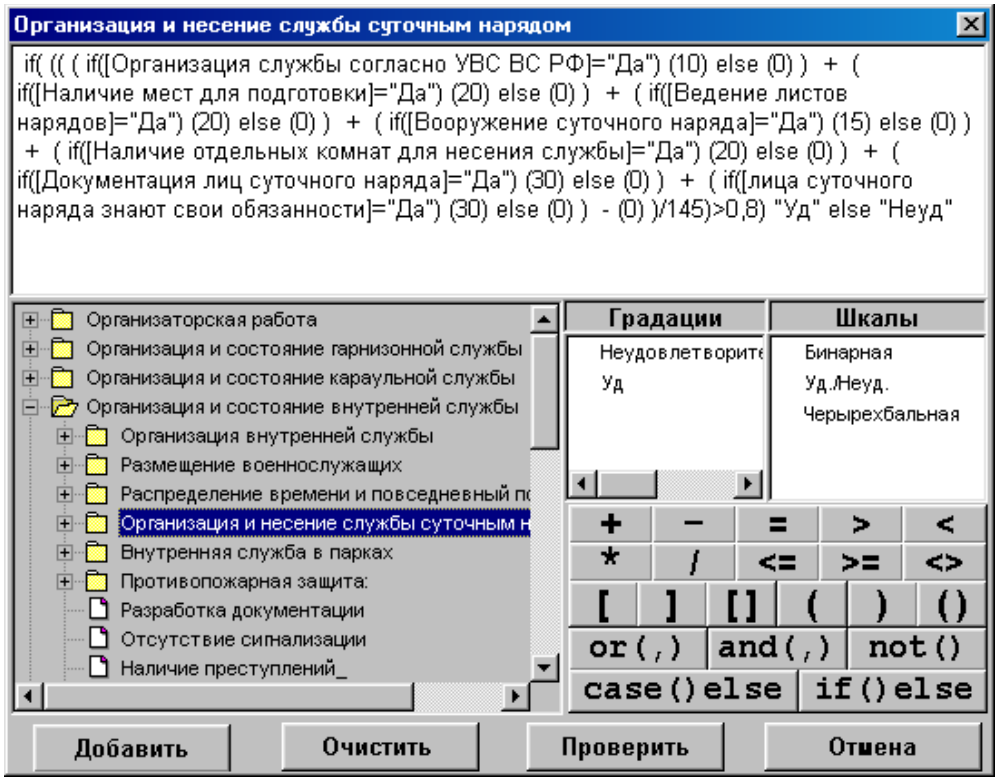


Рис. 2.15. Окно редактора формул.

Специфика предметной области потребовала предусмотреть разграничение уровней доступа пользователям к функциям и пунктам меню. Для этого заводиться несколько групп пользователей и для каждой группы определяются права доступа (см. рисунок 2.16.).

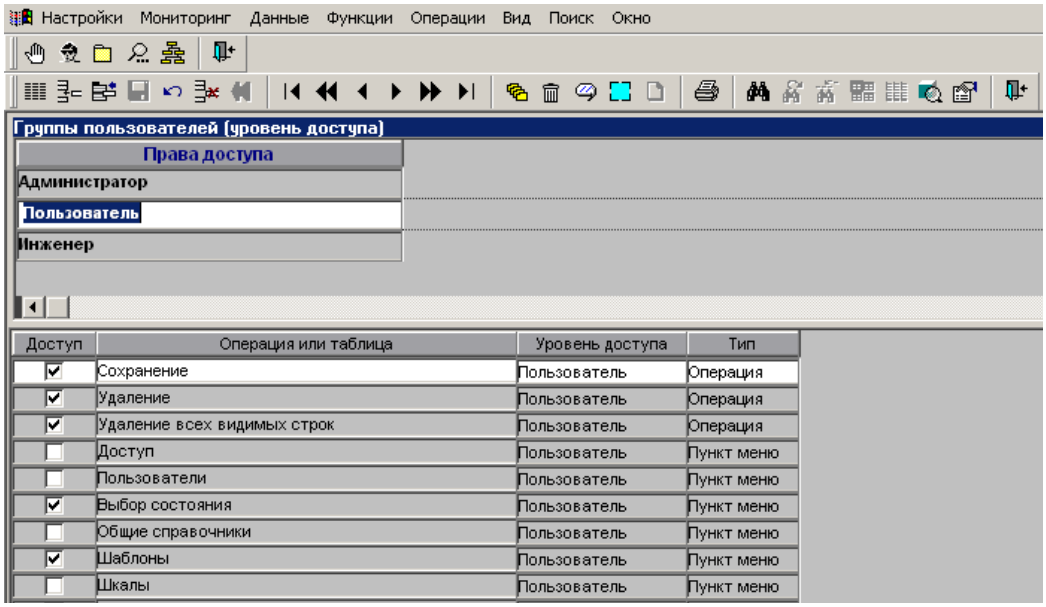


Рис. 2.16. Окно уровней доступа.

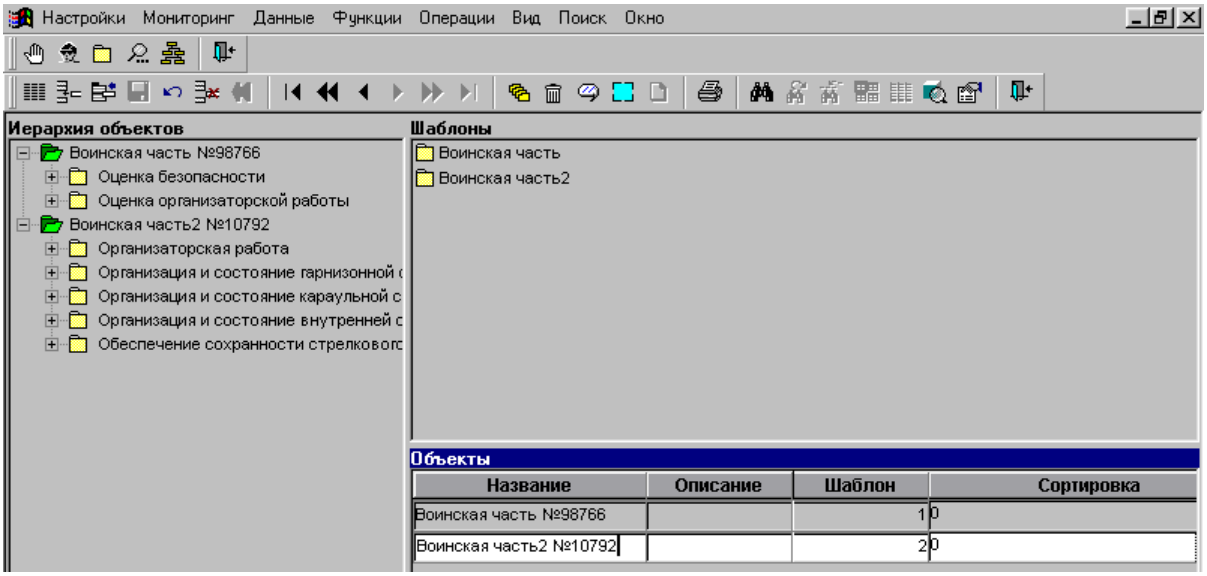


Рис. 2.17. Окно редактирования объектов.

Другой отличительной особенностью АСКУ является окно редактирования иерархии объектов (см. рисунок 2.17.). Структура объекта наследуется от какого-либо шаблона. Создание нового объекта реализуется переносом нужного шаблона в окно объектов. При этом в выделенный объект в качестве подчиненного будет вставлен новый объект, структура которого унаследована от выбранного шаблона (а если ничего не выбрано, новый объект будет определен как новый корневой). Имя нового объекта идентично имени шаблона, но его можно изменить в панели редактирования свойств объектов.

2.2.4. Оценка состояния войсковых частей и отделов

Созданная система апробировалась при решении задачи оценки состояния воинских частей и подразделений по направлениям:

- оценка безопасности военной службы;
- оценка службы войск;
- оценка по боевой подготовке;
- комплексная оценка.

Настройка системы была осуществлена в соответствии с предоставленными методиками. В соответствии с ними было сформировано дерево показателей, которое включало в себя три ветки для оценки соответствующих направлений (оценка безопасности военной службы, оценка службы войск, оценка по боевой подготовке), которые сводятся к комплексной оценке. Для сверки показателей на всех уровнях кроме высшего были использованы арифметические и логические формулы. На высшем уровне используется свертка трех показателей по зависимым предпочтениям (см. рисунок 2.18.).

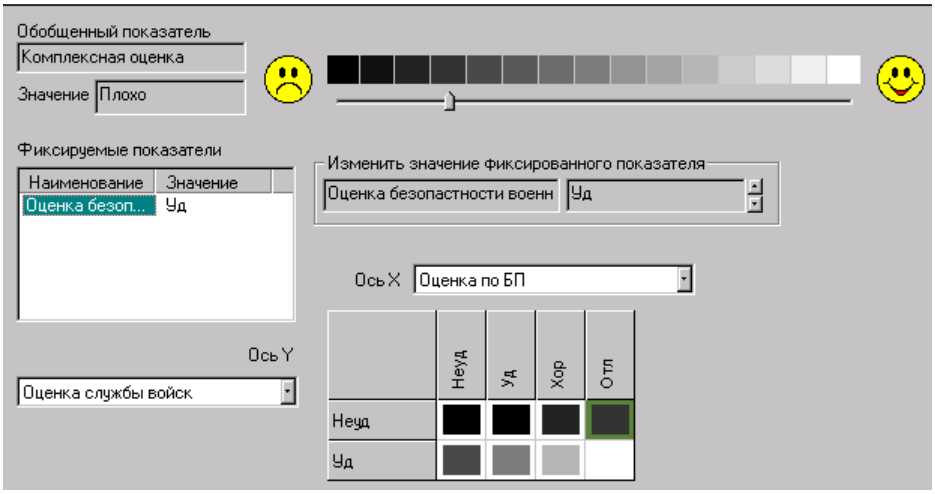


Рис. 2.18. Ввод предпочтений для оценки комплексной части.

Особенностью первых задач оценки безопасности военной службы, оценки службы войск является:

- лексический характер всех исходных показателей с бинарным типом шкалы (возможные значения: «Да» или «Нет»);
- лексический или численный характер обобщенных показателей;
- логический характер вычислений обобщенных показателей с использованием логического условия «Если то» и с последующим нормированием;
- двухбалльная оценка на верхнем уровне с градациями «Уд.» и «Неуд.».

Задачу оценки боевой подготовки отличает следующее:

- все показатели (как исходные, так и обобщенные) являются численными и оцениваются в пятибалльной шкале (5 – отлично, 4 – хорошо, 3 – удовлетворительно, 2 - неудовлетворительно);
- оценка боевой подготовки части осуществляется с учетом организационно-штатной структуры (при назначении оценок по боевой подготовке части учитываются оценки по боевой подготовке отделов в неё входящих);
- формулы, реализующие оценки по боевой подготовке не требуют корректировки при изменении организационно штатной структуры конкретных воинских частей.

Фрагмент результатов расчетов по методике оценки безопасности военной службы для двух воинских частей приведен на рисунке 2.19.

Как видно из рисунка 2.19. оценка состояния первой воинской части лучше по показателю 1.1.2. Причины этого можно увидеть на следующем уровне иерархии, где в частности показатель 1.1.2.1. получил более высокую оценку в баллах для первой части. Это же можно сказать и про показатель 1.1.2.5. Опускаясь по дереву показателей далее можно увидеть, какие исходные показатели привели к лучшей или худшей обобщенной оценке. Так значения показателей 1.1.2.1.1. и 1.1.2.1.2. равняются «Да», и поэтому первая воинская часть получает за каждый из них по пять баллов, отрицательные ответы для второй воинской части не приводят к увеличению её суммарной оценки.

Состояние [Состояние на 26.11.2001]
Упорядочено по показателю [Оценка безопастности военной службы]

Номер в иерархии	Название	32103 Вч	77981 вч
1.1.2.	Обучение личного состава требованиям безопасности военной службы	Уд	Неуд
1.1.2.1.	Занятия в системе	10	0
1.1.2.1.1.	Занятия в системе боевой (командирской) подготовки	Да	Нет
1.1.2.1.2.	Проведение занятий в системе по ОП	Да	Нет
1.1.2.2.	Обучение организованно с серьезными недостатками	Нет	Нет
1.1.2.3.	Наличие отчётных документов_	Да	Да
1.1.2.4.	Обучение осуществляется при изучении учебной дисциплины "Безопасность жизнедеятельности"	Нет	Нет
1.1.2.5.	Осуществление допуска к самостоятельной работе	Уд	Неуд
1.1.2.6.	Осуществление обучения без допуска к самостоятельной работе	Нет	Нет
1.1.2.7.	Отсутствие занятий по СП	0	-25

Рис. 2.19. Оценки показателей нижних уровней.

Введенные предпочтения позволили определить комплексную оценку двух тестовых войсковых частей, которая приведена на рисунке 2.20. Отличие комплексной оценки обуславливается разницей в оценке безопасности военной службы, которая далее может быть развернута вплоть до исходных показателей.

Номер в иерархии	Название	77981 вч	32103 Вч
1.	Комплексная оценка	Недопустимо	Отлично
1.1.	Оценка безопастности военной службы	Неуд	Уд
1.1.1.	организация подготовки безопасности	Неуд	Уд
1.1.1.1.	Инструктажи личного состава по требованиям безопасности в воинской части	Неуд	Уд
1.1.1.1.1.	Инструктажи проводятся бессистемно	Нет	Нет
1.1.1.1.2.	Инструктажи проводятся согласно требованиям директивы	Нет	Да
1.1.1.2.	Обучение личного состава требованиям безопасности военной службы	Неуд	Уд
1.1.1.2.1.	Занятия в системе	0	10
1.1.1.2.1.1.	Занятия в системе боевой (командирской) подготовки	Нет	Да
1.1.1.2.1.2.	Проведение занятий в системе по ОГП	Нет	Да

Лист 1 из 33

Рис. 2.20. Сравнение комплексных оценок воинских частей.

Внесенные в систему методики имеют принципиальный характер, т.к. на этой же основе в дальнейшем могут быть осуществлены и все последующие свертки.

2.3. СППР ГЛОНАСС

На сегодняшний день финансовое состояние и эффективность деятельности промышленной корпорации любого профиля определяется умением ведущих менеджеров компании эффективно организовать управляющие процессы внутри самой корпорации.

Очень часто источниками проблем управления корпорацией являются отсутствие формализованных стратегических целей, недостаточное планирование организационных процессов, недооценка рисков и принятие необоснованных решений. Выбор выгодных краткосрочных оперативных решений может оказаться в противоречии со стратегическими целями и привести к значительным финансовым убыткам впоследствии.

В связи с этим, представляется целесообразным проанализировать функциональные задачи подсистем, обеспечивающих функционирование системы ГЛОНАСС, рассмотреть жизненный цикл системы и оценить возможности создания эффективной системы поддержки стратегических и оперативных решений в различных эшелонах управления корпорацией.

2.3.1. Состав системы ГЛОНАСС

Глобальная навигационная спутниковая система (ГЛОНАСС) является одной из федеральных целевых программ Российской Федерации и предназначена для определения местоположения, скорости движения, а также точного времени морских, воздушных, сухопутных и других видов потребителей [154].

Система ГЛОНАСС состоит из трех подсистем [1, 2]:

- подсистемы космических аппаратов (ПКА);
- подсистемы контроля и управления (ПКУ);

- навигационной аппаратуры потребителей (НАП) / аппаратуры спутниковой навигации (АСН).

Подсистема космических аппаратов системы ГЛОНАСС состоит из 24 навигационных космических аппаратов (НКА), находящихся на круговых орбитах высотой 19100 км, наклонением 64,8° и периодом обращения 11 часов 15 минут в трех орбитальных плоскостях. Орбитальные плоскости разнесены по долготе на 120°. В каждой орбитальной плоскости размещаются по 8 НКА с равномерным сдвигом по аргументу широты 45°. Кроме этого, в разных плоскостях положения НКА из разных плоскостей сдвинуты относительно друг друга по аргументу широты на 15°.

Такая конфигурация ПКА позволяет обеспечить непрерывное и глобальное покрытие земной поверхности и околоземного пространства навигационным полем.

По плану развития этой программы на период с 2012 по 2020 годы предусматривается увеличение орбитальной группировки до 30 космических аппаратов. Это обеспечит гарантированное получение сигналов навигационной аппаратурой, увеличит целостность и точность координат. Предусматривается также использование функциональных дополнений, включающих системы мониторинга, системы дифференциальной коррекции и беззапросные технологии. Дифференциальные корректирующие станции и системы на их основе позволят получить повышенную — сантиметровую — точность, которая необходима для мониторинга инженерных сооружений, башен, мостов, железных дорог и др.

ПКУ состоит из Центра управления системой ГЛОНАСС и сети станций измерения, управления и контроля, рассредоточенных по всей территории России. В задачи ПКУ входит контроль правильности функционирования ПКА, непрерывное уточнение параметров орбит и выдача на НКА временных программ, команд управления и навигационной информации.

НАП состоит из навигационных приемников и устройств обработки, предназначенных для приема навигационных сигналов НКА и вычисления собственных координат, скорости и времени.

В настоящее время всего несколько отечественных предприятий серийно выпускают ГЛОНАСС/GPS OEM-модули. Это продукция предприятий ОАО «Ижевский радиозавод», ЗАО «КБ НАВИС», ОАО «РИРВ», ФГУП НИИМА «Прогресс».

Помимо перечисленных выше предприятий, еще ряд компаний ведут разработку навигационного оборудования ГЛОНАСС/GPS. В их числе такие предприятия телекоммуникационного рынка как АФК «Система», SPIRIT Telecom и Javad GNSS.

Несмотря на перспективы развития рынка навигационной аппаратуры, в настоящее время основными потребителями оборудования, в основе которого лежат ГЛОНАСС/GPS-приемники, являются силовые ведомства, службы МЧС, кадастровые службы [137].

Регулируемый рынок, по оценкам Федерального агентства по промышленности, составит более $120 млн. в год. Потенциальная емкость этого сегмента оценивается в 700–750 тыс. комплектов.

Большим и перспективным рынком для ГЛОНАСС/GPS-оборудования являются перевозчики. Правительство РФ приняло постановление от 9 июня 2005 года № 365, предписывающее применять аппаратуру ГЛОНАСС/GPS на всех морских и речных судах, а также железнодорожном и автомобильном транспорте, если он используется для перевозки пассажиров, специальных и опасных грузов. По заявлению министра транспорта РФ Игоря Левитина, сделанного на втором Международном транспортном форуме «Транспорт России-2008», к 2012 году всем российским и зарубежным перевозчикам, находящимся в РФ, будут выдвинуты требования по наличию установленной навигационной аппаратуры ГЛОНАСС/GPS на борту.

Активно развивается проект оснащения ГЛОНАСС/GPS-приемниками поездов для более эффективного управления их движением, а также применения ГЛОНАСС/GPS-маячков в контейнерных перевозках. ОАО «Российские железные дороги», согласно «Стратегии развития железнодорожного транспорта в России до 2030 года», в период с 2007 по 2011 год планировалось оснастить навигационной аппаратурой около 2,5 тыс. пассажирских локомотивов и свыше 17 тыс. грузовых, включая магистральные электровозы и тепловозы, а также маневровые тепловозы.

Таможенные службы также собираются применять ГЛОНАСС для борьбы с простоями грузов на таможнях. Основная идея — оформлять таможенные грузы не на границах, а в специально созданных терминалах внутри государства, транспорт же, следующий к этим терминалам, оснащать навигационным оборудованием.

Большая цена, незавершенность развертывания космической группировки ГЛОНАСС, достаточно высокое энергопотребление приемников, несомненно, являются сдерживающими факторами развития этого рынка.

Вместе с тем, следует предполагать, что с развитием аппаратной базы приемников, отечественные аппараты будут пользоваться все большим спросом, при условии конкурентоспособности по отношению цена/ качество.

Поскольку рынок услуг ГЛОНАСС относится к рынку высокотехнологичной продукции, то следует также ожидать появления нового типа услуг в этой области.

Если учесть, что подсистема космических аппаратов, включает в себя не только сами космические аппараты, но и комплекс предприятий по их производству, испытаниям, отладке и запуску, то, анализируя укрупненную структуру системы ГЛОНАСС, можно отметить, что каждая подсистема состоит из совокупности достаточно самостоятельных предприятий,

имеющих кроме корпоративных и свои собственные цели и интересы. Систему ГЛОНАСС в целом можно рассматривать как многоуровневую иерархическую систему с несколькими центрами управления, и не всегда однозначно определенными связями между ними.

Следует отметить, что в силу тесного переплетения деятельности государства и отдельных предприятий система ГЛОНАСС представляет собой сложную слабоструктурированную систему, обладающую рядом таких свойств, как уникальность, трудная формализуемость, высокая степень неопределенности, динамичность, невозможность полного описания и др. С учетом отмеченных особенностей исследование процессов функционирования и управления такой системой целесообразно переводить на методы компьютерной поддержки решений.

2.3.2. Сравнение DSS/UTES и ExpertChoice в задаче выбора навигационных приемников

Для того чтобы оценить сильные и слабые стороны различных систем и методов поддержки принятия решений рассмотрим их применение на конкретной задаче.

Одними из ключевых факторов успеха навигационной системы являются её потребительские качества. Потребительские качества определяются, в том числе, и качеством используемых навигационных приемников. Рассмотрим задачу выбора навигационных приемников поддерживающих ГЛОНАСС. Перечень таких приемников представлен на сайте http://www.aggf.ru/catalog/razdel/sod.php.

Предположим, что стоит задача выбора навигационных приемников индивидуального пользования. В качестве СППР, которые помогут решить эту задачу, рассмотрим 2 системы: ExpertChoice 11.5. и DSS/UTES 2.0.

Работа с обеими системами начинается с определения показателей оценки. На рисунках 2.21. и 2.22. представлены основные окна систем в режиме ввода показателей.

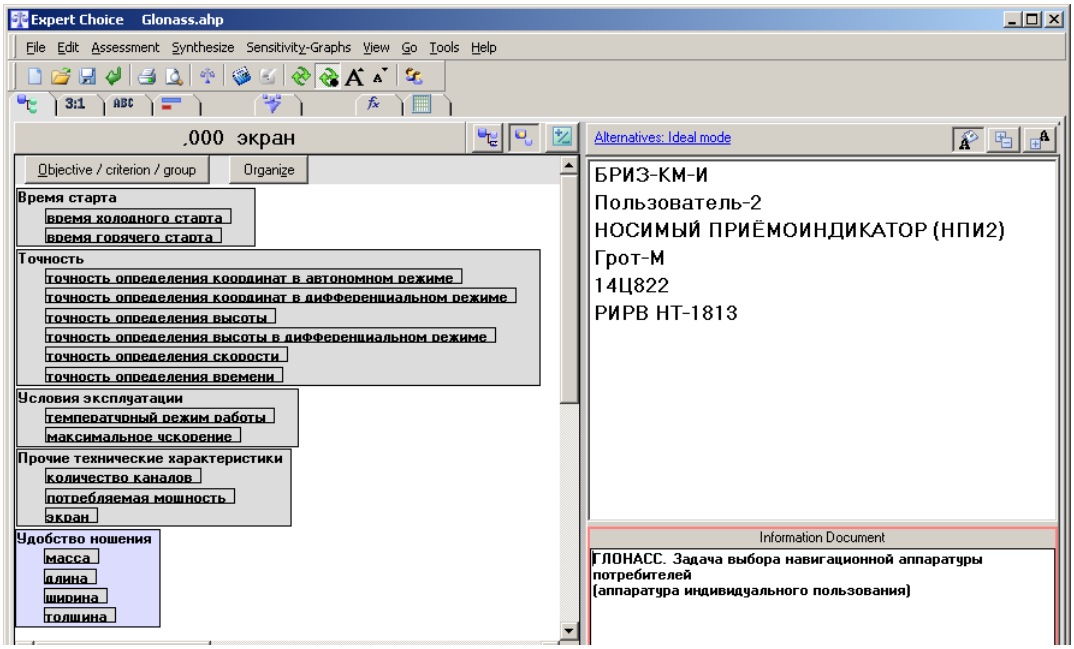


Рис 2.21. Структура показателей в ExpertChoice.

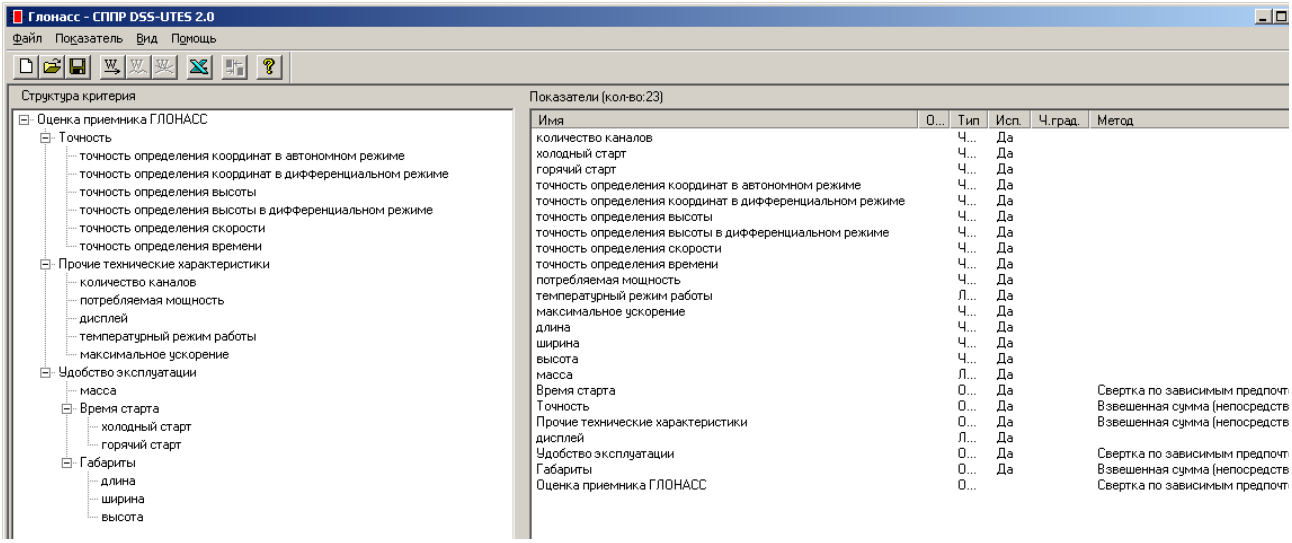


Рис. 2.22. Дерево показателей в DSS/UTES.

Этап формирования пространства показателей весьма принципиален и играет определяющую роль в процессе принятия решений. Все дальнейшие операции по поиску наилучших решений и ранжированию альтернатив будут производиться в границах этого пространства. Выполнение этого этапа вынуждает пользователя четко сформулировать свое понимание задачи и цели, которую он хочет достичь. Диалог с системой должен помогать ему в этом.

К сожалению ExpertChoice не обладает русскоязычным интерфейсом. В некоторых окнах русские названия показателей и альтернатив не отображаются корректно. Интерфейс DSS/UTES русский. ExpertChoice позволяет представить иерархию показателей в виде вложений прямоугольников друг в друга. Создание иерархии показателей облегчается путем

простого перетаскивания прямоугольников друг в друга. Обе системы допускают представление в виде стандартного элемента управления типа дерево.

При структуризации показателей в методе анализа иерархий желательно включать в один узел не больше 7 показателей. Иначе их будет трудно сравнивать. В DSS/UTES все зависит от выбранного метода свертки. Для зависимых предпочтений желательно, чтобы на одном уровне было не больше 4 показателей.

Применительно к задаче выбора навигационного приемника ГЛОНАСС в DSS/UTES на верхнем уровне предложены следующие показатели:

- точность;
- удобство эксплуатации;
- прочие технические характеристики.

После того как пространство показателей сформировано, в DSS/UTES пользователь приступает к построению своей функции предпочтений. На рисунке 2.23. приведена экранная форма, соответствующая этапу формирования функции предпочтений. Показаны раскрашенное двумерное сечение ФП, перечень пассивных показателей с указаниям "замороженных" значений и палитра полутонов. Все полутона перенумерованы, что облегчает пользователю процесс раскрашивания. Эту же нумерацию можно рассматривать как множество возможных значений функции предпочтения. Соответственно, чем больше полутонов содержит палитра, тем выше разрешающая способность системы по предпочтениям. В частности, для того, чтобы различать по предпочтениям все возможные комбинации значений показателей, их число должно равняться числу полутонов палитры. Но практически требуемое (или используемое) количество полутонов заметно меньше, т.к. существует большое число равных по предпочтениям комбинаций значений показателей. В частности, равными по предпочтениям являются все комбинации, в которых хотя бы один показатель имеет недопустимое значение.

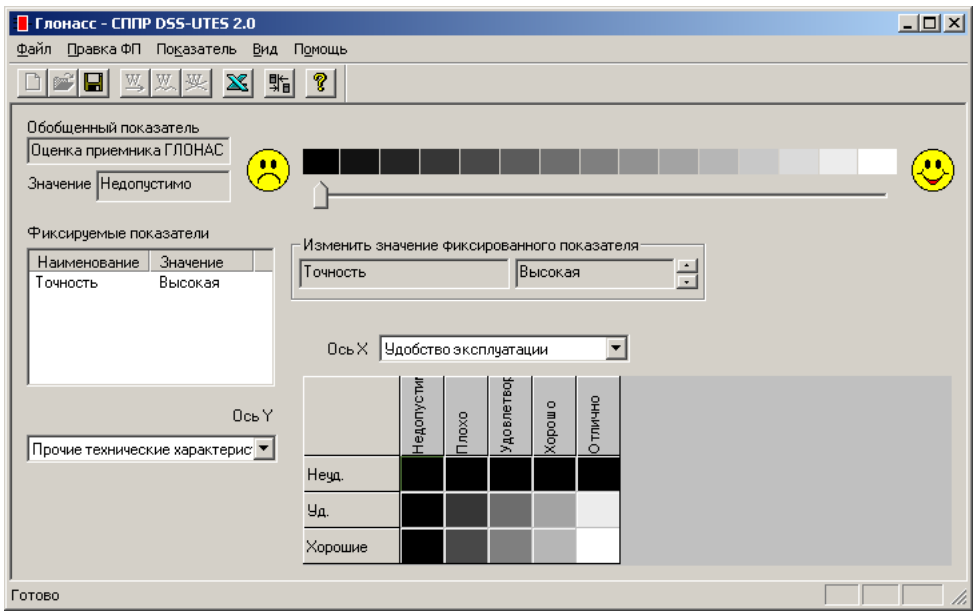


Рис. 2.23. Предпочтения.

Показатель точность приемника ГЛОНАСС определяется следующими характеристиками:

- точность определения координат в автономном режиме;
- точность определения координат в дифференциальном режиме;
- точность определения высоты;
- точность определения высоты в дифференциальном режиме;
- точность определения скорости;
- точность определения времени.

Для свертки всех этих характеристик в обобщенную оценку точности можно воспользоваться взвешенной суммой. Эта процедура хороша тем, что она прозрачна и отражает систему ценностей лица, ее осуществляющего. Однако она возможна только, если размерность векторного критерия невелика. В противном случае она может оказаться практически не выполнимой.

На рисунке 2.24. показано, как ввести непосредственные веса показателей в DSS/UTES. Аналогично непосредственные веса можно вводить и в ExpertChoice (см. рисунок 2.25.).

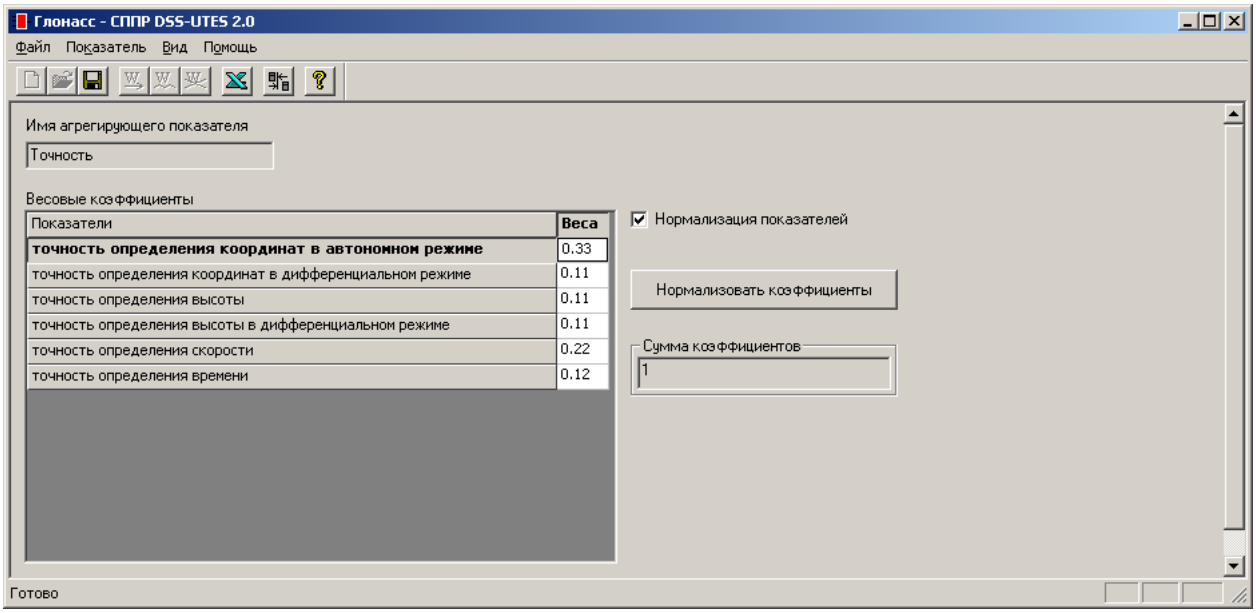


Рис. 2.24. Взвешенная сумма в DSS/UTES.

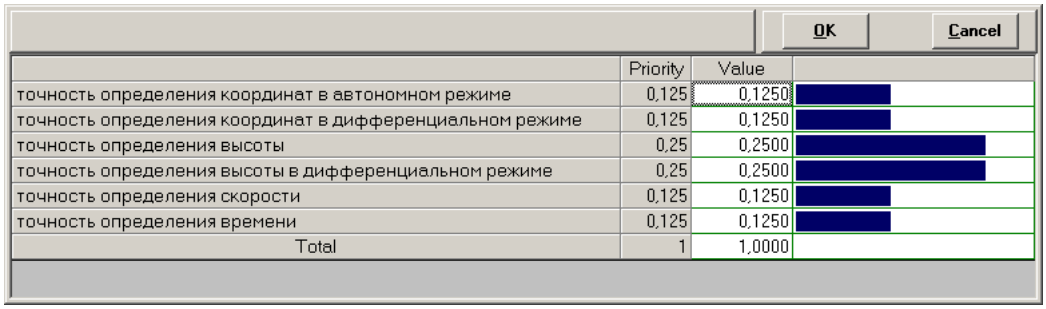


Рис. 2.25. Непосредственное назначение весов в ExpertChoice.

Для определения весов для большого количества показателей обычно используется процедура парных сравнений. При непосредственном назначении элементов матрицы парных сравнений эксперты или ЛПР могут проявить непоследовательность суждений при сопоставлении отдельных пар, что приведет к несогласованности элементов матрицы и возникновению петель нетранзитивности. В этом случае необходима коррекция сделанных назначений. На рисунках 2.26. и 2.27. показано как вводятся парные сравнения на верхнем уровне дерева показателей в ExpertChoice в графическом представлении и в девятибалльной шкале соответственно.

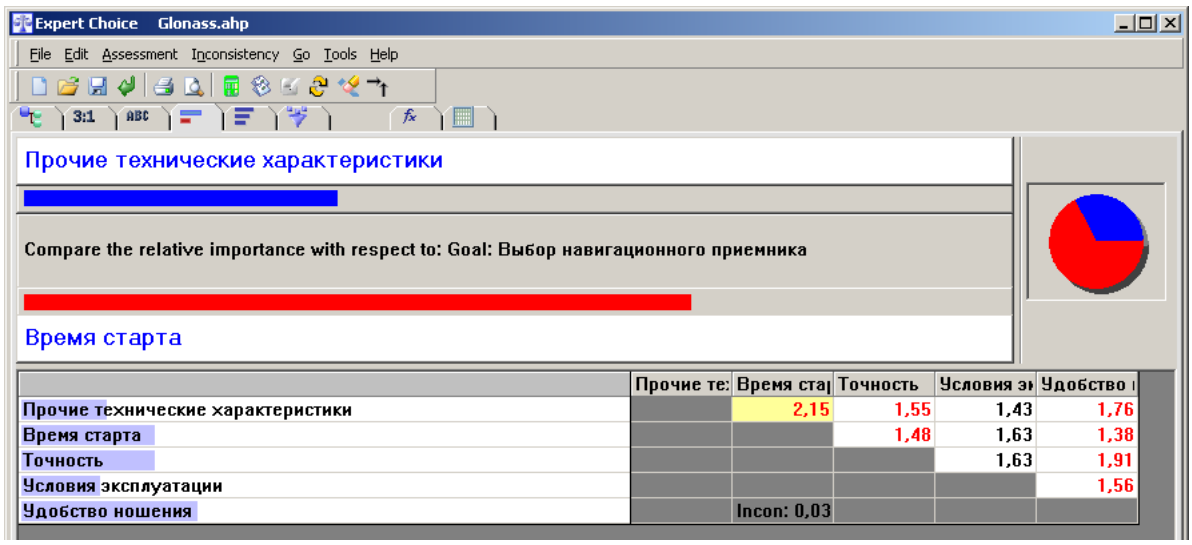


Рис. 2.26. Парные сравнения показателей верхнего уровня.

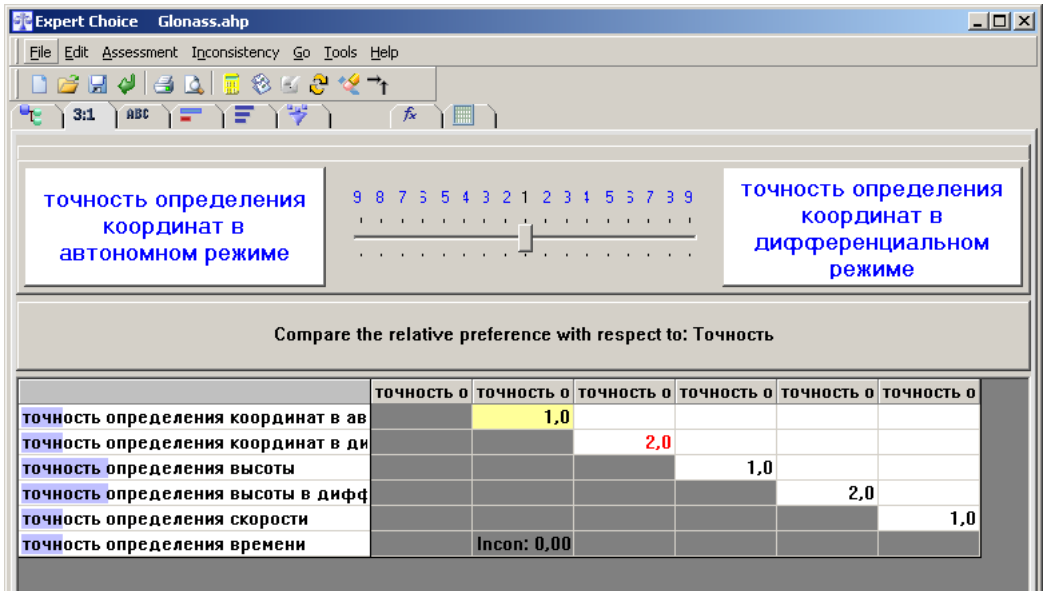


Рис. 2.27. Парные сравнения в бальной шкале.

В ExpertChoice альтернативы задаются в том же файле, что показатели, система предлагает их ввести на ранних этапах работы. Результат их интегральной оценки показан на рисунке 2.28.

В DSS/UTES ввод альтернатив организован в отдельном файле формата XLS, его можно заполнять сразу после ввода критерия, а можно и позже. Таким образом, в DSS/UTES не требуется дополнительная работа ЛПР при появлении новых альтернатив, их ранжирование происходит в автоматическом режиме. На рисунке 2.29. показано, как вводятся альтернативы в DSS/UTES.

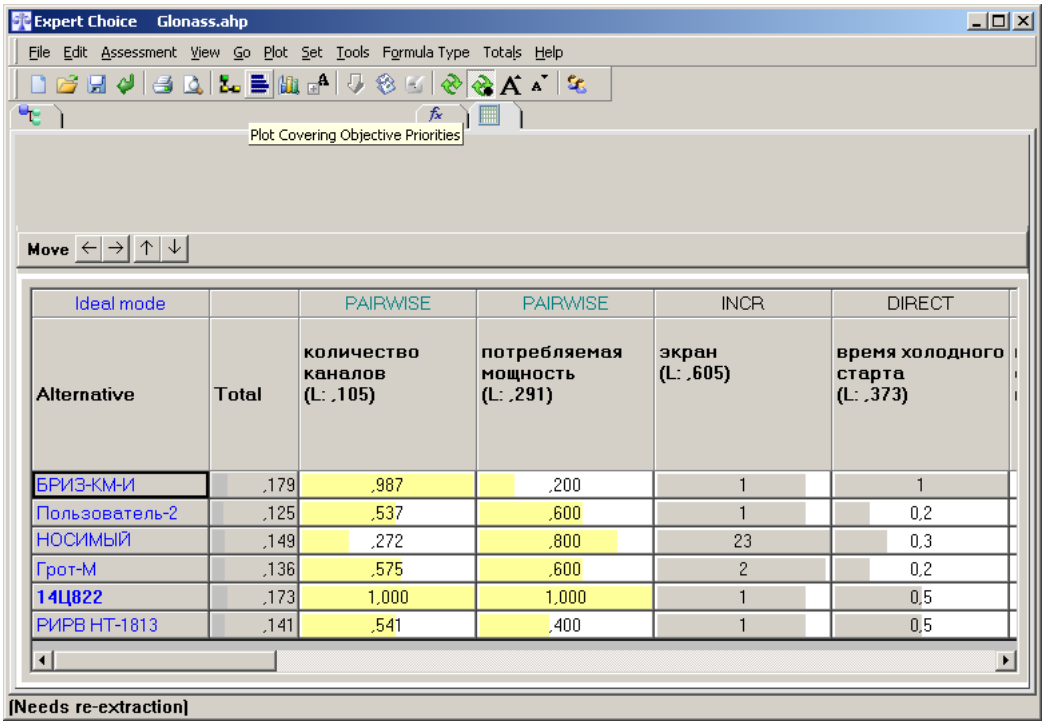


Рис. 2.28. Результат оценки альтернатив.

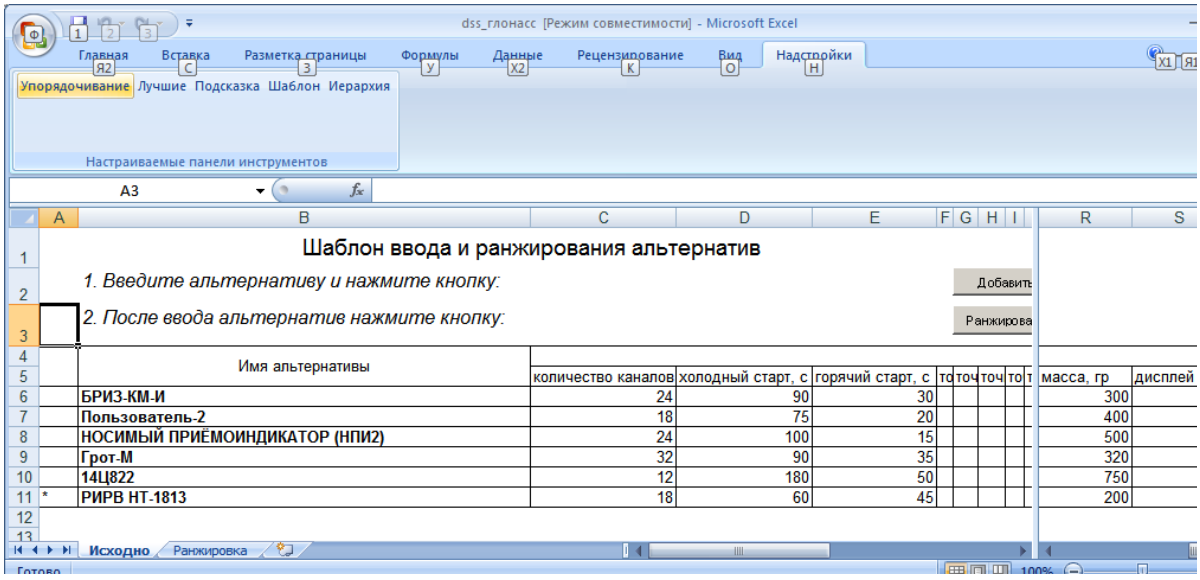


Рис. 2.29. Интерфейс ввода и ранжирования альтернатив в DSS/UTES.

По результатам работы с рассмотренными системами можно сделать следующие выводы:

- DSS/UTES обеспечивает работу как с лексическими показателями без оцифровки. ExpertChoice требует перевода лексических показателей в числа;
- изначальное дерево показателей в ExpertChoice вводить удобней;
- в ExpertChoice нет русскоязычного интерфейса. Русские названия показателей и альтернатив отображаются некорректно;
- DSS/UTES поддерживает ввод зависимых предпочтений которого нет в ExpertChoice;

- в DSS/UTES ввод альтернатив отделен от ввода предпочтений, а ExpertChoice при добавлении новых альтернатив требует их новых парных сравнений.

Таким образом, можно сделать вывод, что DSS/UTES позволяет достичь большей гибкости и эффективности в решении задач выбора приемников ГЛОНАСС, хотя и требуются некоторые улучшения интерфейса.

2.3.3. Программное обеспечение СППР ГЛОНАСС

Программное обеспечение СППР ГЛОНАСС создавалось с использованием свободного системного программного обеспечения (ПО) [24, 28].

Применительно к отечественным разработкам в космической отрасли использование свободного системного ПО позволяет минимизировать риски:

- риск наличия в программе вредоносного и «шпионского» кода. Подобный код в свободном ПО легко обнаружить;
- риск прекращения поддержки и развития системы. Систему всегда можно дорабатывать самим.

Поэтому СППР для нужд космической отрасли целесообразно создавать на базе свободного системного программного обеспечения. Причем оно может быть как отечественным, так и зарубежным, это не принципиально, главное, чтобы оно обладало всеми свободами, указанными в определении свободного ПО, и в России были специалисты, которые могли бы его поддерживать и дорабатывать [135].

В качестве СУБД для СППР ГЛОНАСС использовался PostgreSQL. Сильными сторонами PostgreSQL считаются: поддержка БД практически неограниченного размера, мощные и надёжные механизмы транзакций и репликации, наследование, легкая расширяемость.

Целесообразно, чтобы ПО СППР ГЛОНАСС работало под управлением свободной операционной системы. Здесь на сегодняшний день доминируют различные версии Linux систем. Безусловно, для конечного пользователя работа под Linux может оказаться некомфортной, это является одной из причин для организации Web-интерфейса к системе. При этом на серверах должен работать Linux, а на клиенте - любая удобная пользователю операционная система.

Для разработки СППР ГЛОНАСС использовался язык программирования Ruby и набор базовых классов Rails. Ruby обладает простым и лаконичным синтаксисом. К достоинствам Ruby и Rails следует отнести наличие их реализации в открытых исходных кодах и возможность свободно модифицировать эти исходные коды. Таким образом обеспечивается независимость от зарубежных производителей системного ПО.

Возможности СППР ГЛОНАСС продемонстрированы на примере задачи оценки навигационных приемников (см. рисунок 2.30.).

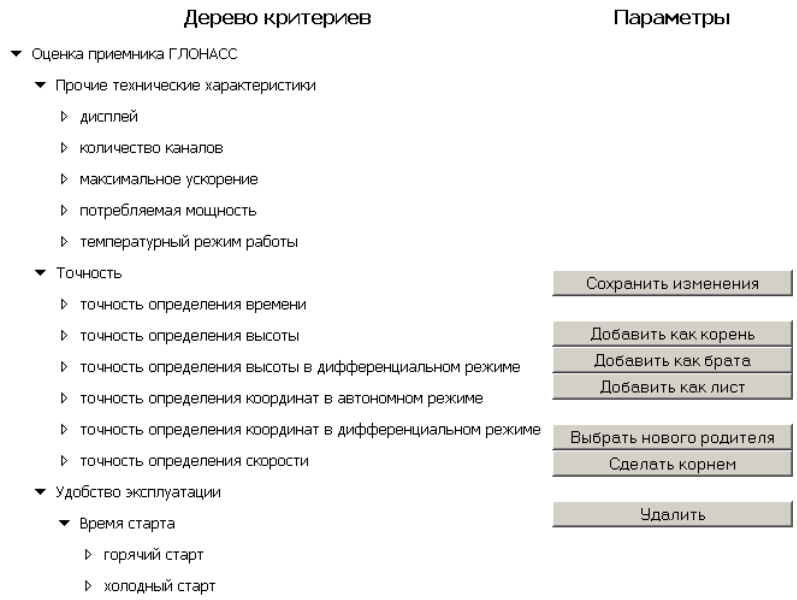


Рис. 2.30. Дерево критериев в задаче выбора навигационных приемников в СППР ГЛОНАСС.

2.4. Выводы по второй главе

- Рассмотренные задачи ракетно-космической отрасли являются ресурсоемкими и высокоразмерными. Для их решения требуется применение развитого математического, алгоритмического и информационного обеспечения, позволяющего решать многокритериальные задачи, осуществлять многоуровневую декомпозицию векторного критерия, для систем, имеющих иерархическую структуру. Требуется открытый характер программного обеспечения.
- Существенные трудозатраты при создании рассмотренных систем связаны с разработкой индивидуального информационного обеспечения. Поэтому важной задачей является разработка универсальной структуры такого обеспечения.
- Настройки на конкретные задачи поддержки решений ракетно-космической отрасли позволяет осуществлять СППР DSS/UTES. Существенным недостатком СППР DSS/UTES является высокая трудоемкость построения ФП. Поэтому актуальной задачей является разработка более совершенного метода построения ФП.

Глава 3. Разработка нового метода решения задач с высокой размерностью векторного критерия

3.1. Постановка задачи

Решаемые в работе практические задачи из ракетно-космической отрасли, характеризуются десятками критериев, имеют зависимости по предпочтениям между критериями, содержат лексические критерии, оцифровка которых связана со значительными трудностями [58].

Функция предпочтений (ФП) в СППР DSS/UTES:

- позволяет учесть зависимости по предпочтениям;
- проста и легко понимается ЛПР;
- позволяет практически мгновенно ранжировать произвольное количество альтернатив, после того как предпочтения заданы.

Однако методу ФП в СППР DSS/UTES присущи следующие недостатки:

- трудоемкость формирования ФП;
- неразличимость вариантов внутри ячейки, образованной градациями;
- актуальность проблемы по разбиению шкалы на градации;
- невозможность строить ФП более чем для 5 критериев.

Исходя из сказанного выше, видна актуальность задачи разработки нового гибридного метода построения ФП [59]. Метод создан с целью решения следующих задач:

- обеспечить ЛПР инструментарием для качественных суждений о предпочтительности альтернатив в задачах с высокой размерностью векторного критерия (под качественными понимаются суждения в порядковой шкале, когда ЛПР может сказать, что одно решение лучше другого, ЛПР может указать, что один критерий важнее другого, но при этом затруднительно получить числовые оценки степени превосходства);
- выявить предпочтения ЛПР во всем критериальном пространстве, чтобы дальнейшая оценка конкретных альтернатив проходила в автоматическом режиме (таким образом, процесс выбора и ранжирования альтернатив происходит быстро и появляется возможность проводить процедуру поиска решений без привлечения ЛПР, но с учетом его предпочтений);
- учесть зависимости между компонентами векторного критерия по предпочтениям (в результате можно устранить ситуации, когда альтернативы с неприемлемыми для

ЛПР оценками по одному критерию, получают высокую интегральную оценку за счет других критериев);

- обеспечить различимость альтернатив в случае, когда значения критериев подвергаются искусственной дискретизации с целью замены непрерывных шкал на балльные оценки.

Метод носит названия гибридного, так как сочетает элементы качественных методов (функции предпочтений, качественные важности критериев, метод ЗАПРОС) и количественных методов (взвешенная сумма, идеальная точка) выявления суждений ЛПР.

Дадим описание сути предлагаемого гибридного метода в общем виде.

Допустим, пространство критериев имеет размерность ***n***, равную размерности векторного критерия оптимальности. Оси координат соответствуют частным критериям оптимальности. Далее, каждой точке этого пространства ставится в соответствие значение некоторой функции предпочтений, отражающей систему ценностей ЛПР о том, какие решения лучше, а какие хуже.

ЛПР выделяет в пространстве критериев непересекающиеся области. Практика показывает, что ЛПР, исходя из своих знаний и опыта, легко выделяет области неудовлетворительных решений, области удовлетворительных решений, области хороших решений для достаточно небольшого (5-16) количества областей. Для применения метода необходимы знания об области допустимых значений каждого из критериев, но не требуется, чтобы все точки критериального пространства были допустимыми, знания о предпочтениях в недопустимых точках просто не используются. Для построения областей ось каждого критерия разбивается на интервалы, в результате критериальное пространство разбивается на отдельные ячейки прямоугольной формы. На рисунке 3.1. показан пример критериального пространства для случая 2 критериев, градации критериев выделены пунктирными линиями, пересечения пунктирных линий образуют ячейки.

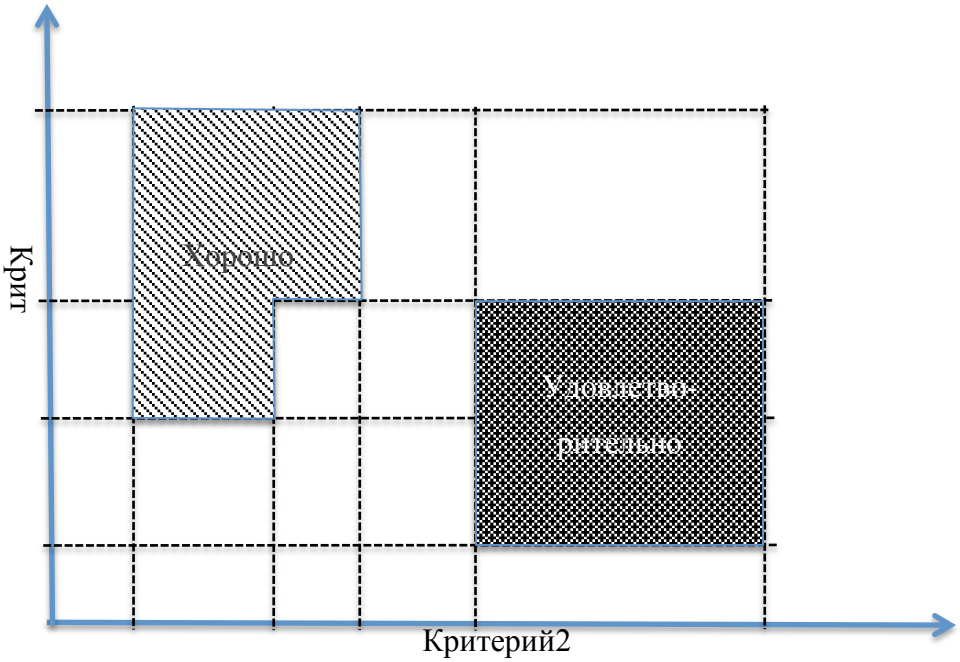


Рис. 3.1. Пример пользовательских областей предпочтений.

Пользователь указывает одну или несколько таких ячеек и сообщает СППР, что в этой области уровень предпочтений имеет некоторое значение в заданной лексической шкале: хорошо, удовлетворительно, неудовлетворительно и т.д. На рисунке 3.1. показаны две области, которые указал пользователь. Детальный алгоритм по назначению уровней предпочтений для n-мерного случая будет показан ниже на рисунке 3.2. после формализации метода. Уровень предпочтений позволяет определить, что решения в одной области лучше решений в другой области, но не говорит, на сколько лучше. Таким образом, уровень предпочтений выражается в порядковой шкале. При этом должна соблюдаться транзитивность предпочтений. Проверка транзитивности суждений ЛПР выполняется СППР автоматически и будет показана ниже.

В практических задачах ранжирования альтернатив часто несколько недоминируемых альтернатив попадают в одну область. Возникает вопрос об их сопоставлении. В этом случае предлагается воспользоваться формальным (количественным) методом сопоставления альтернатив внутри заданной области. Такими формальными методами могут быть: взвешенная сумма, мультипликативная свертка, идеальная точка, свертка Гермейера, расстояние Чебышева. Применение формальных методов в локальной области основано на следующем эвристическом соображении: зависимости по предпочтениям и существенные нелинейности проявляются только при больших изменениях значений критериев. Под большими изменениями мы понимаем такие, которые приводят к переходу из одной области предпочтений в другую. В небольшой области недостатки формальных методов, такие как взаимная компенсация

критериев, влияние масштабирующих коэффициентов проявляют себя несущественно для пользователя.

3.2. Формализация задачи и алгоритмы решения

Для математической формализации метода, дадим следующие обозначения:

i – номер критерия,

i=1..n, где *n* – число критериев.

С целью удобства задания областей, которые будем упорядочивать по предпочтениям, разобьем значения критериев на градации:

t_{ij} - *j*-я градация *i*-го критерия.

Число градаций у критериев может быть различным:

j=1.. q_i, где q_i – число градаций i-го критерия.

Для лексических критериев в качестве таких градаций выступают слова (термы). Для числовых критериев градация - это интервал значений. Интервал указывается в виде правой и левой границы.

По каждому критерию предпочтения должны быть монотонны или иметь один экстремум (идеальное значение – например, комфортная температура в помещении). Градации должны быть отсортированы в порядке увеличения предпочтительности:

$$\forall i, j : t_{ij+1} \succ t_{ij} .$$

Полное пространство всех возможных комбинаций значений градаций критериев задается как декартово произведение:

$$A = \{t_{11}, t_{12}, ..., t_{1q_1}\} \otimes \{t_{21}, t_{22}, ..., t_{2q_2}\} \otimes ... \otimes \{t_{n1}, t_{n2}, ..., t_{nq_n}\}.$$

Мощность этого множества вычисляется по формуле:

$$Q = |A| = \prod_{i=1}^{n} q_i .$$

В большинстве практических задач значение *Q* велико, например, для 7 критериев с 5 градациями: $Q=5^7=78125$. Задавать уровни предпочтения для всех элементов *A* на практике сложно. Поэтому в качестве области, для которой определяется уровень предпочтения, будем рассматривать объединение некоторых элементов множества *A*.

Определим *множества* M_k, где *k=1..K*. Множество M_k представляет собой множество точек критериального пространства в определенной пользователем прямоугольной области. Для каждого M_k по каждому из критериев определяются градации, которые в него входят:

m_{ik} – множество номеров градаций i-го критерия, которые входят в k-е множество M_k. Множества M_k – это подмножества множества *A*.

Рассмотрим произвольную точку пространства критериев: $X=\{x_1, x_2, \ldots, x_n\}$, где x_i – значение i-го критерия, которое обязательно попадает в одну из градаций $\exists j(x_i \in t_{ij})$. Принадлежность X к множеству M_k определяется предикатом:

$$W(X,k) = \mathop{\&}_{i=1}^{n}\left(\mathop{\vee}_{j\in m_{ik}} (x_i \in t_{ij}) \right).$$

Для каждого уровня предпочтений P определяются входящие в него области с заданным уровнем предпочтений $P(M_k)$. Множество значений P – это целые числа, чем больше число, тем предпочтительнее соответствующий аргумент функции. Над значениями функции $P(\ldots)$ нельзя производить арифметических операций сложения, вычитания и прочих, допустимы только операции сравнения. На рисунке 3.2. в нотации UML показано, как ЛПР вводит и редактирует множества M_k.

СППР проверяет, что все определенные пользователем множества M_k с различным уровнем предпочтений должны отличаться хотя бы по одному критерию. Для этого, после появлении новой области или изменении существующей области автоматически поверяется выполнение выражения:

$$\forall k_1, k_2 (P(M_{k_1}) \neq P(M_{k_2})) \rightarrow (\exists i(m_{ik_1} \cap m_{ik_2} = \varnothing)) .$$

Если выражение не выполняется, то пользователь должен скорректировать свои назначения предпочтений или границы множеств M_k. Если множества M_k имеют равный уровень предпочтений, то допустимо их пересечение.

Для алгоритма показанного на рисунке 3.2. в качестве входа выступают критерии и списки градаций для каждого критерия. Пользователь задает лексическую шкалу уровней предпочтений и для каждого из уровней предпочтений определяет одно или несколько множеств M_k, указывая градации по критериям. После ввода каждого множества, проверяется что оно не пересекается с остальными. На выходе получается совокупность множеств M_k заданных через $\{m_{ik}\}$, для каждого из которых задан уровень предпочтений $P(M_k)$. При сообщении о пересечении должна выдаваться подсказка ЛПР: необходимо, чтобы хотя бы по одному из критериев градации не было общих градаций.

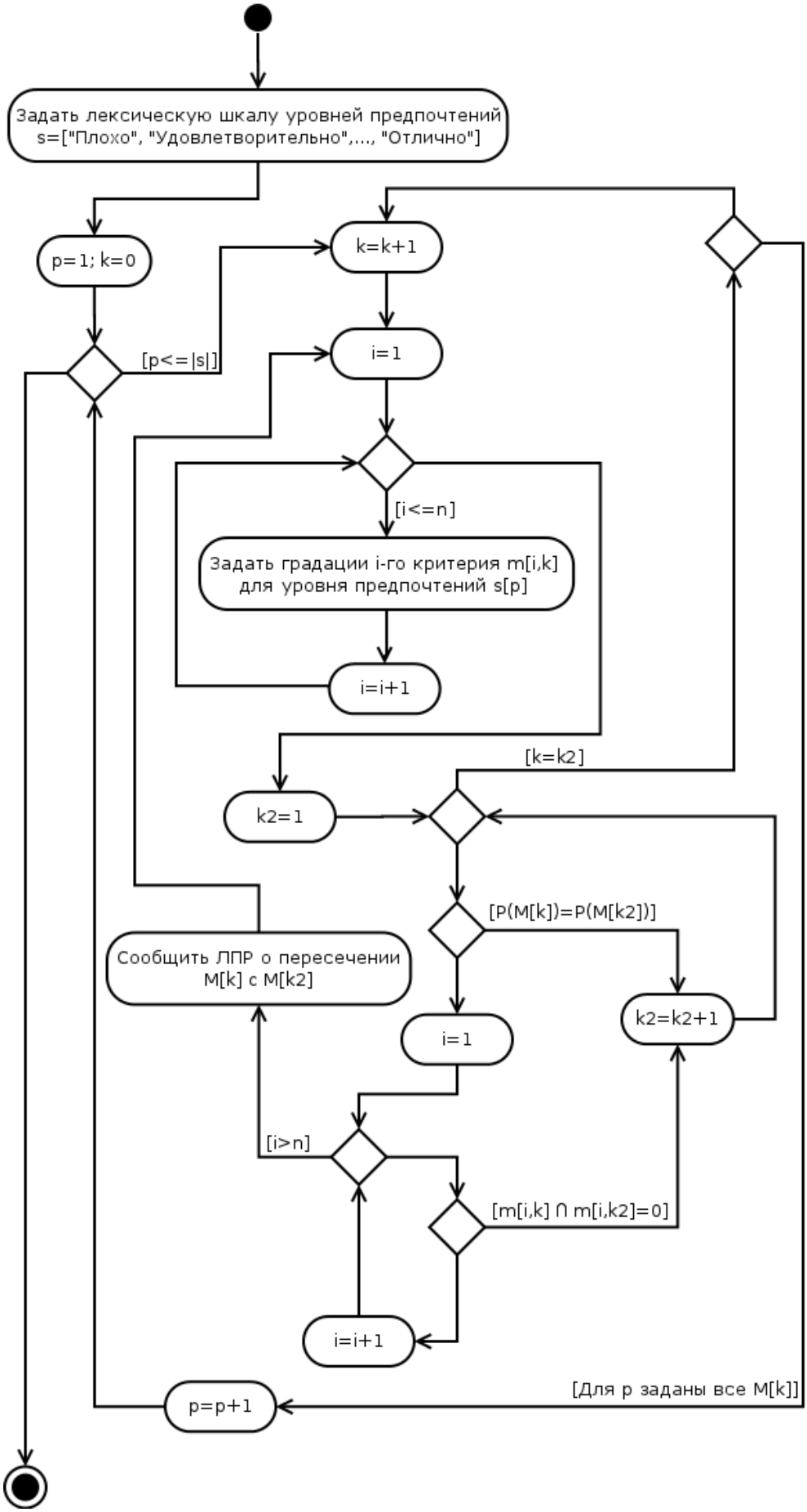


Рис. 3.2. Алгоритм ввода и редактирования M_k и уровней предпочтений.

В приложении 1 приведена полная трассировка работы гибридного метода выявления предпочтений.

Так как объединение всех множеств M_k в общем случае не составляет всего множества A, то необходимо обеспечить возможность определения уровня предпочтений, для любой комбинации значений градаций, которые не попали ни в одно из M_k. Назовем такие комбинации градаций ***ячейками*** и обозначим их как

$$T_l = \{t_{1g_{1l}}, t_{2g_{2l}}, ..., t_{ng_{nl}}\},$$

где g_{il} – номер градации i-го критерия для l-й ячейки, $l=1..L$.

Нужно найти все комбинации градаций, которые не попадают ни в одно из множеств M_k. Условие непопадания во все множества M_k запишется как:

$$\forall l \forall k \exists i (g_{il} \notin m_{ik}).$$

На рисунке 3.3. в нотации UML показан алгоритм поиска таких комбинаций градаций. В нем анализируются все комбинации градаций. Для этого используется вспомогательный массив s ={ s[i] – номер градации i-го критерия}. Входом для алгоритма являются $\{m_{ik}\}$ на выходе формируются T_l. Алгоритм работает по методу прямого полного перебора всех комбинаций градаций по всем критериям (текущая комбинация хранится в s). Каждая текущая комбинация проверяется на принадлежность всем M_k, если для всех M_k принадлежность не найдена, то s добавляется в $\{T_l\}$.

Определим общее количество таких ячеек L. Для этого нужно вычесть из общего количества комбинаций градаций значений критериев Q число комбинаций, которые уже покрыты всеми множествами M_k. Если M_k не пересекаются, то количество комбинаций, которые они покрывают, можно сложить:

$$L = Q - \sum_{k=1}^{K} \prod_{i=1}^{n} |m_{ik}|.$$

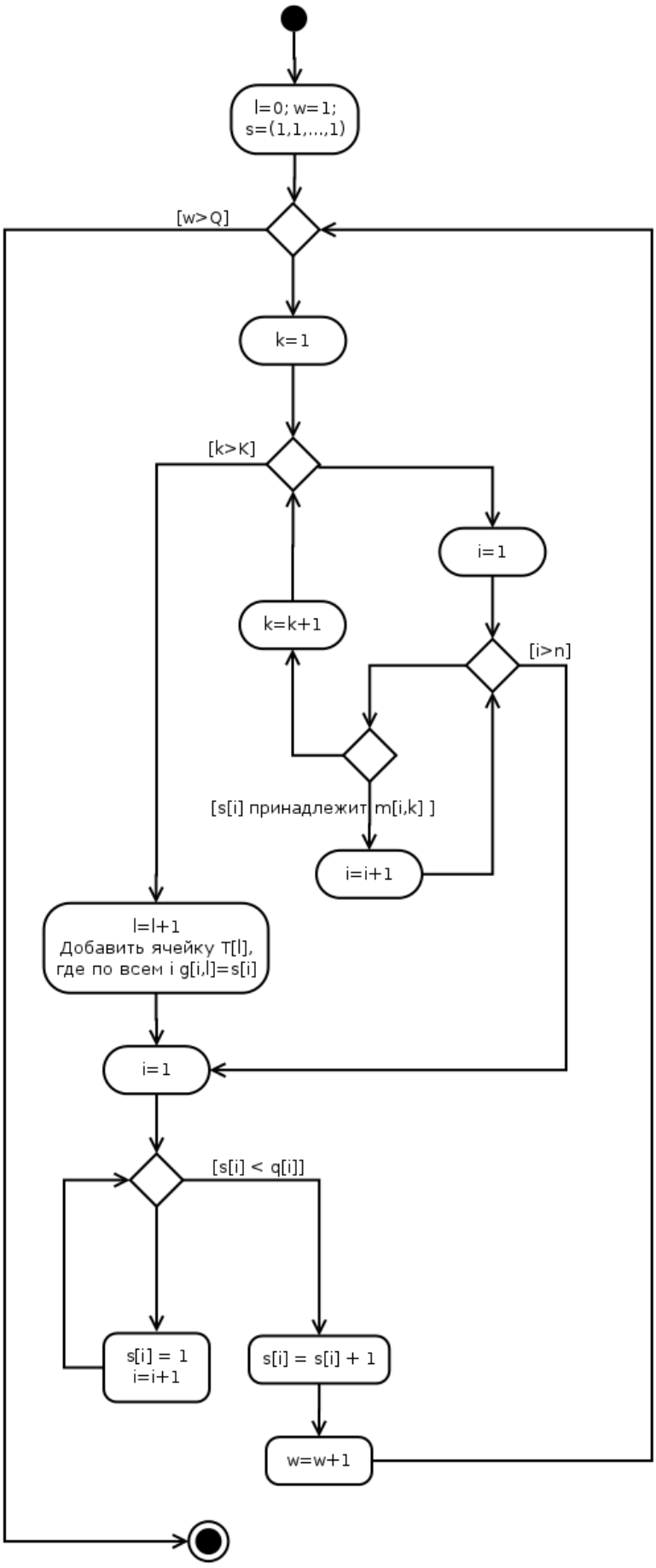


Рис. 3.3. Алгоритм определения ячеек T_l.

Учитывая, что градации отсортированы в порядке увеличения предпочтительности, применим к ячейкам T_l и множествам M_k условие оптимальности по Парето. В результате возникают отношения предпочтения одних ячеек над другими:

$$(\forall i(g_{il_1} \geq g_{il_2}) \& \exists i(g_{il_1} > g_{il_2})) \rightarrow T_{l_1} \succ T_{l_2}.$$

А также отношения предпочтения ячеек над множествами:

$$\forall i(g_{il} \geq \max_{j \in m_{ik}} j) \& \exists i(g_{il} > \max_{j \in m_{ik}} j) \rightarrow T_l \succ M_k.$$

и отношения предпочтения множеств над ячейками:

$$\forall i(\min_{j \in m_{ik}} j \geq g_{il}) \& \exists i(\min_{j \in m_{ik}} j > g_{il}) \rightarrow M_k \succ T_l.$$

Кроме того, существует отношение доминирования между всеми множествами M_k. Оно определяется уровнями предпочтений, заданными пользователем:

$$\forall k_1 \forall k_2 (P(M_{k_1}) > P(M_{k_2}) \rightarrow M_{k_1} \succ M_{k_2}).$$

Для некоторых ячеек можно определить предпочтения между ячейками методом качественного учета важностей Подиновского, если выполняются следующие условия:

1. Для критериев установлено отношение предпочтения, например, критерий i важнее критерия i+1, при этом не утверждается на сколько важнее.
2. Критерии являются однородными, то есть для них используется одна и та же шкала, например, пятибалльная шкала экспертных оценок от 1 до 5.

Между ячейками, которые получаются друг из друга перестановкой номеров градаций по двум критериям в случае их однородности, получается отношение предпочтения по правилу Подиновского:

$$\exists i_1 \exists i_2 (i_1 < i_2) \& (\forall i(i \neq i_1 \& i \neq i_2) \rightarrow (g_{il_1} = g_{il_2})) \& (g_{i_1 l_1} > g_{i_1 l_2}) \&$$
$$\& (g_{i_2 l_1} < g_{i_2 l_2}) \& (g_{i_1 l_1} = g_{i_2 l_2}) \& (g_{i_2 l_1} = g_{i_1 l_2}) \rightarrow T_{l_1} \succ T_{l_2}$$

Это правило говорит о том, что в случае, если одна ячейка получается из другой перестановкой значений двух критериев, то из этих двух ячеек предпочтительней та, где большее значение указано у более важного критерия. Упорядочивание критериев по важности осуществляется пользователем.

Применяя указанные правила доминирования ко всем ячейкам и множествам, можно построить ориентированный граф доминирования ячеек T_l и множеств M_k. Вершины этого графа будут соответствовать ячейкам и множествам, а направленные дуги будут говорить об отношении доминирования. Граф нужно проверить на наличие циклов. И если таковые будут найдены, то следует указать на это ЛПР. Цикл в графе говорит о наличии нетранзитивности в суждениях ЛПР. На рисунке 3.4. в нотации UML показан алгоритм корректировки

предпочтений в случае циклов. Необходимо для каждого цикла определить множества $M_{k,}$ которые в него входят, и сообщить пользователю, что необходимо ввести корректировки уровней предпочтений или изменить сами множества, например, назначить M[k2] уровень предпочтений меньше, чем уровень предпочтений M[k1]. В практических задачах возникновение цикла – редкое явление. На реальных исходных данных цикл не возникал, чтобы проверить корректность алгоритма обнаружения циклов в приложении 1 были сформированы специальные исходные данные, цикл устраняется за одну итерацию.

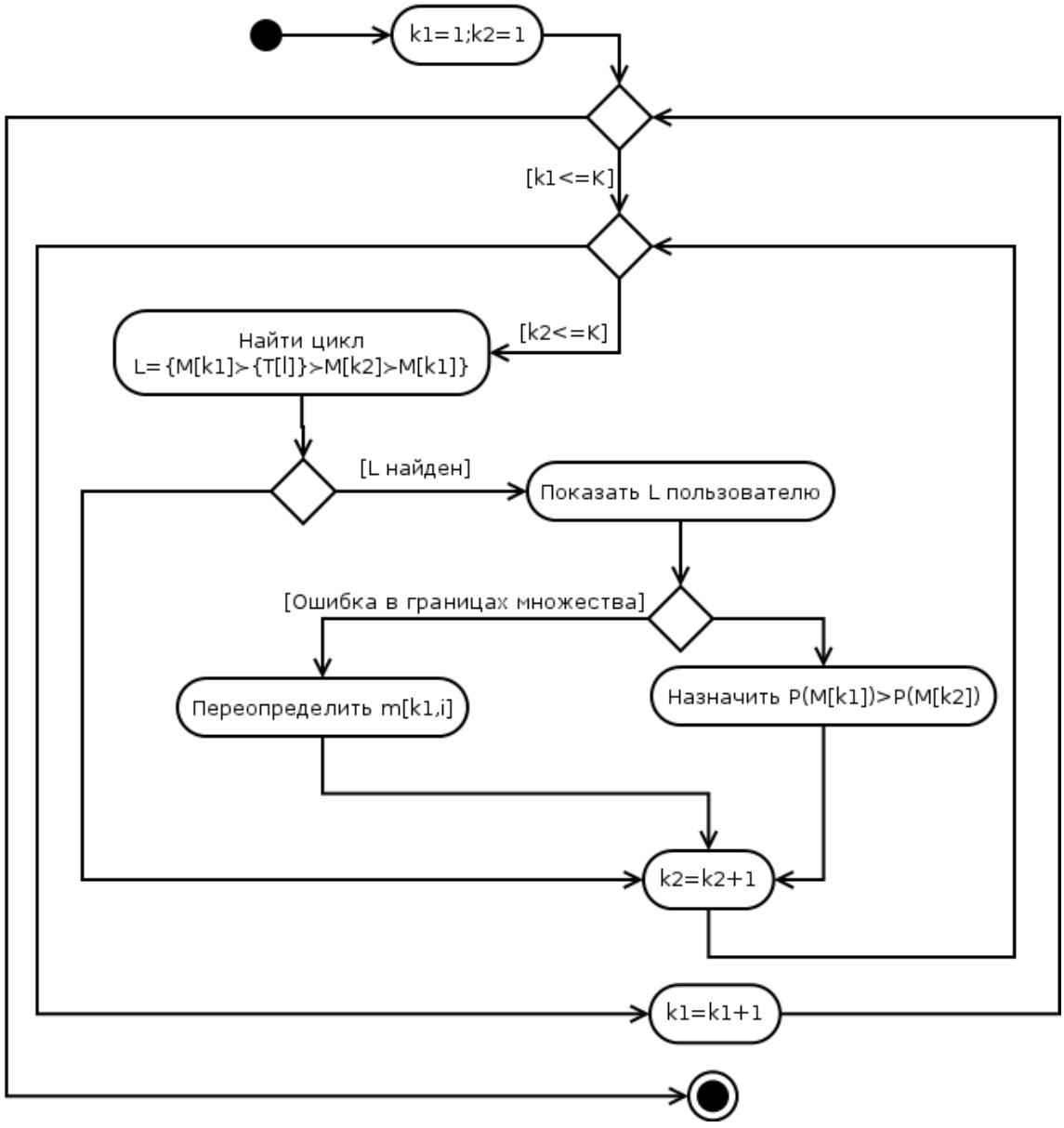


Рис. 3.4. Алгоритм корректировки предпочтений в случае циклов.

После того как будет установлено отсутствие циклов, следует использовать алгоритм анализа графа, который академик Ларичев назвал «разборка» графа. В этом алгоритме определяются значения уровней предпочтений для M_k и T_l:

1. $p = L+K$ – это максимальный уровень предпочтений, который возможен в случае, если все M_k и T_l различимы по предпочтительности.
2. Определить множество недоминируемых вершин M_k и T_l.
3. Присвоить им уровень предпочтений p.
4. Исключить из графа недоминируемые вершины.
5. Если граф пуст, то завершение алгоритма.
6. $p = p - 1$.
7. Переход на шаг 2.

После применения данного алгоритма, некоторые вершины попадут на один уровень предпочтений. Если все вершины будут различимы, то в конце работы алгоритма переменная p будет равна нулю. Если все вершины неразличимы, то p будет равна $L+K$. Определим Z - синтетический показатель уровня неразличимости на полном множестве:

$$Z = \frac{p}{L+K} \cdot 100\% .$$

Следует заметить, что чем детальней пользователь задаст свои предпочтения, тем меньше будет это число. Если пользователя не устраивает результат работы метода (например, не устраивают ранги альтернатив полученные в результате ранжирования альтернатив указанным методом), то следует указать пользователю на необходимость понижения значения Z. Понизить значение Z можно путем ввода новых множеств M_k.

Обычно ячейка или множество включает более одного возможного значения критерия. Для возможности сопоставления различных векторов пространства критериев, входящих в одно множество или в одну ячейку, необходимо добиться изменения итоговой оценки альтернативы при изменении критерия в рамках одного уровня предпочтений. Для этого предлагается вычислять итоговое значение предпочтений по формуле:

$Y(X) = p(X) + D(X)$, где

$p(X)$ – целочисленный уровень предпочтений, советующей области M_k или ячейки T_l, в которую попало значение X, определенный вышеописанным алгоритмом разбора графа;

$D(X)$ – оценка, полученная количественным методом свертки векторного критерия, вычисленная в рамках той области (множества или ячейки), куда попало значение вектора X и нормированная в пределах от 0 до 1.

Функцию $Y(X)$, вычисляющую итоговое значение предпочтений, назовем *гибридной функцией предпочтений* (ГФП).

Итоговая оценка альтернативы, полученная указанным методом, состоит из целой части, отражающей качественные суждения пользователя, и дробной части, отражающей применение

количественных методов, с целью повышения чувствительности и возможности оптимизации на непрерывном множестве.

На рисунке 3.5. показан общий алгоритм гибридного метода формирования функции предпочтений. Данный гибридный метод, с одной стороны, позволяет существенно упростить процедуру построения функции предпочтений, задавая ее подробно не на всем критериальном пространстве. С другой стороны, использовать преимущества, представляемые количественными методами свертки векторных критериев и требующие, как правило, меньшего объема информации при их использовании, для более детального сопоставления альтернатив, находящихся в одной области решений, одинаковой по предпочтению с точки зрения ЛПР.

Далее, в пятой главе будет показан результат построения ГФП в задаче оценки значимости космических экспериментов проводимых на борту МКС. Циклов не возникло (сходимость за одну итерацию). В задаче предпочтения задавались для 6 критериев, СППР DSS/UTES нельзя задать функцию предпочтений на таком пространстве и пришлось бы прибегнуть к агрегированию показателей. В СППР DSS/UTES задание предпочтений заняло около 4 часов. ГФП была задана за 1 час. В приложении 1 приведены результаты расчетов. Все альтернативы с уникальными комбинациями значений критериев получили уникальные ранги при ранжировании.

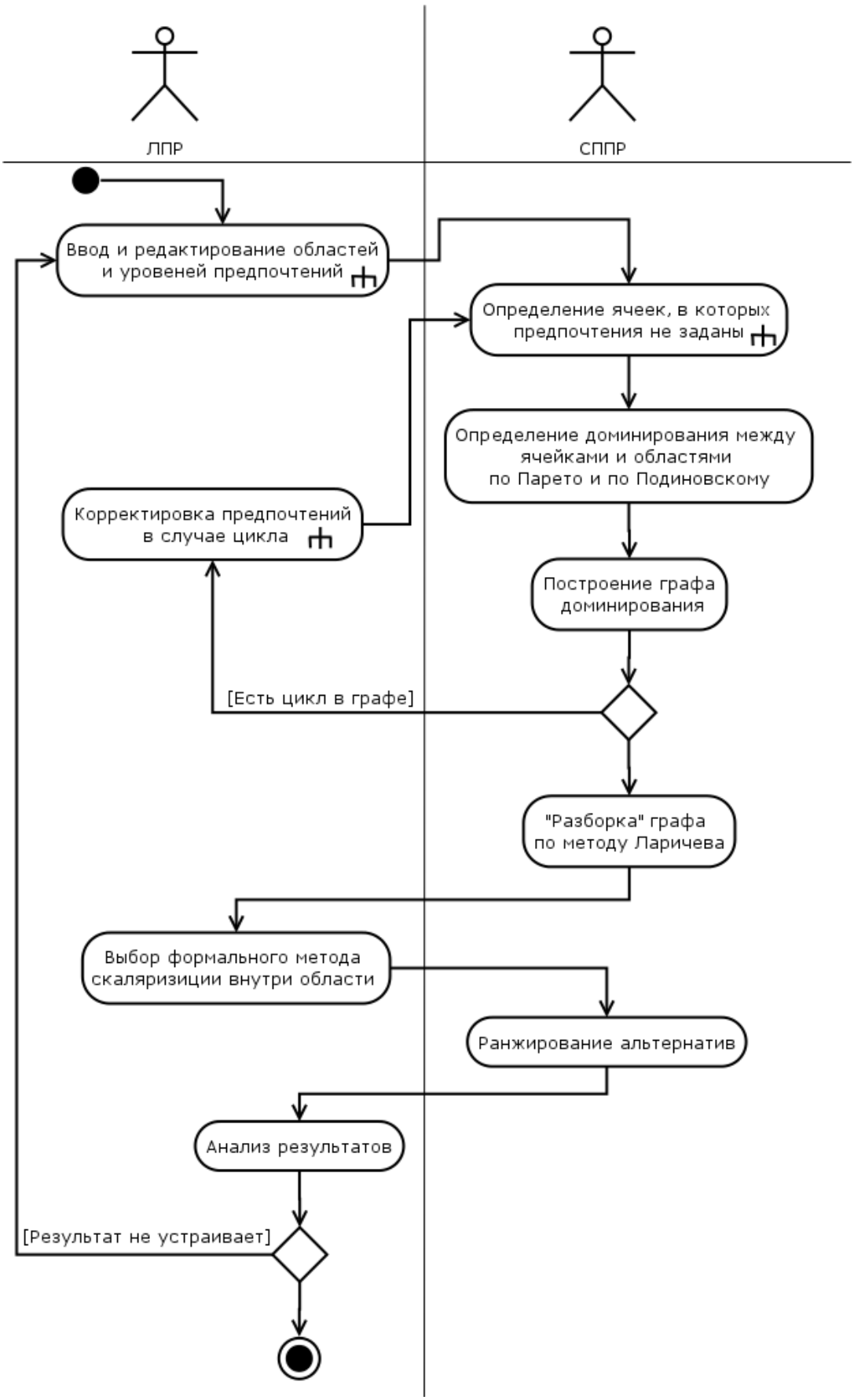


Рис. 3.5. Алгоритм формирования гибридной функции предпочтений.

3.3. Дискретизация шкал критериев

Рассмотрим процедуру построения многоуровневого дерева критериев. При этом подходе показатели объединяются в группы, и каждая из них характеризуется своим показателем называемым *обобщенным.* Формирование групп должно происходить с участием или под контролем ЛПР. Для каждого обобщенного показателя задается правило его вычисления (решающее правило). В качестве решающего правила может выступать формальный метод свертки или процедура свертки по предпочтениям, в результате которой вычисляется «промежуточная ФП». Процедуру объединения показателей в группы назовем агрегированием. Оно выполняется по смысловому признаку, т.е. агрегированные показатели несут обобщенную информацию о каком-либо аспекте решаемой задачи, детально определяемом показателями, вошедшими в его состав. В результате агрегирования происходит переход к векторному критерию меньшей размерности. Процедура агрегирования может повторяться многократно. В итоге можно получить дерево показателей, в корне которого находится один обобщенный показатель [23, 145, 148].

Дерево агрегирования во многом напоминает иерархию из метода анализа иерархий. Отличие заключается в том, что для свертки показателей, которые могут быть как числовыми, так и лексическими, в первую очередь, используется свертка по предпочтениям. При необходимости используется не только взвешенная сумма, но и другие методы свертки.

Древовидную иерархическую структуру агрегирования показателей можно представить следующим образом. На нижнем уровне иерархии (листва дерева) образуется $N1$ групп, по q_{1i} элементов в каждой, где q_{1i}-число показателей в i-ой группе “листвы” (q_{1i} =3 или 4), n – число показателей в «листве», а число $N1$ находится в пределах: $]n/\max(q_{1i})[\le N1 \le]n/\min(q_{1i})[$. (поскольку n/q_I не всегда целое число, то оператор]…[означает округление до ближайшего целого большего или равного операнду). На следующем (более высоком) уровне: q_2 - число показателей в группах, а роль листвы исполняют N1 групп первого уровня и т.д. В заключение процедуры в корне дерева может остаться один показатель.

При агрегировании показателей, мы получаем высокую размерность шкал обобщенных показателей. Например, даны 4 показателя и каждый из них характеризуется 5 градациями, чтобы различать все комбинации их значений требуется шкала с 5^4=625 градациями. Назначение предпочтений на таких шкалах затруднительно, поэтому в работе предложена процедура дискретизации шкал обобщенных показателей. Она содержит две подзадачи. Первая - выбор числа градаций, и вторая - разбиение шкалы на градации. Шкалу полученную в результате такой процедуры назовем «загрубленной». При решении первой подзадачи

пользователь, на основе содержательных соображений, сам назначает требуемую точность оценки. Для решении второй подзадачи предлагаются три способа:

1. Разбиение на равные интервалы. В информационном плане, как будет показано далее, этот метод не всегда целесообразен.
2. Предоставление пользователю возможности самому формировать интервалы разбиения на основе содержательного анализа. Неудобство заключается в том, что пользователю системы придется тратить достаточно много времени на разбиение и согласование шкал.
3. Разбиение шкалы обобщенного показателя на основе минимизации информационных потерь или максимизации энтропии показателя.

Очевидно, что дискретизация шкал приводит к потерям информации. Наиболее полная информация о векторном критерии содержится на уровне исходных показателей. Потери возрастают от листвы к корню дерева агрегирования. Оценить потери информации, возникающие в процессе агрегирования показателей, можно с помощью энтропии [71, 120]. Как известно, количество информации измеряется разностью энтропий до и после получения сообщения. В нашем случае энтропия показателя после получения сообщения равна нулю и количество информации в сообщении определяется энтропией показателя до его получения. Поэтому осуществлять разбиение шкалы обобщенного показателя на градации желательно так, чтобы энтропия показателя была максимальной. А это приводит к идее разбиения на такие интервалы, в которых сообщения появлялись бы с одинаковыми вероятностями. Такой подход известен и применялся для оцифровки непрерывных шкал измерений приборов [4].

В рассматриваемой задаче имеется исходная порядковая шкала уровней предпочтений с большим количеством градаций и из нее нужно получить шкалу предпочтений с меньшим количеством градаций с минимальными информационными потерями. Это приводит к оптимизационной задаче, которая решается для каждого обобщенного показателя независимо от уровня агрегирования.

Ее формальная постановка имеет вид:

$$\max_{u_1,u_2,\ldots,u_{m^*}} I^* = -\sum_{r=1}^{m^*}\Big(\sum_{k=\sum_{l=1}^{r-1}u_l+1}^{\sum_{l=1}^{r}u_l} p_k\Big)\log_2\Big(\sum_{k=\sum_{l=1}^{r-1}u_l+1}^{\sum_{l=1}^{r}u_l} p_k\Big),$$

при ограничениях:

$$\sum_{r=1}^{m^*} u_r = m,$$

$u_r>0$, $r=1..m^*$, где

u_r – оптимизационные переменные, которые показывают, сколько градаций исходной шкалы включается в r-ю градацию «загрубленной» шкалы,

p_k – вероятность k-го состояния в исходной шкале (ниже приведены соображения по получению этих величин),

m^* - число градаций в «загрубленной» шкале (может варьироваться, некоторые соображения по выбору этого числа приведены ниже),

m – число градаций в исходной шкале (задается ЛПР),

$r=1..m^*$ - номер градации в «загрубленной» шкале,

$k=1..m$ – номер градации в исходной шкале,

$\sum_{l=1}^{r-1} u_l + 1$ - индекс первой градации исходной шкалы, включенной в r-ю градацию «загрубленной» шкалы,

$\sum_{l=1}^{r} u_l$ - индекс последней градации исходной шкалы, включенной в r-ю градацию «загрубленной» шкалы.

«Узким местом» такой постановки задачи является необходимость знания p_k. К сожалению, нет удовлетворяющего практическим требованиям, простого способа получения вероятностного распределения значений векторного критерия. Тем не менее, некоторое приближенное представление об этом распределении можно получить [43, 44, 131]. В процессе эксплуатации СППР оценки p_k могут быть вычислены на основе статистических наблюдений:

$p_k =$ m/n, где m – число альтернатив у которых значение показателя попадало в k-ю градацию, n – общее число оцениваемых альтернатив.

Если система только начинает функционировать, то можно обратится к субъективным вероятностям, основанным на опросе экспертов с применением, например, метода эталонных лотерей.

Поставленная оптимизационная задача относится к классу задач дискретного динамического программирования. Для её решения необходимо составить функцию состояния (функцию Беллмана) $\Lambda_r(\xi)$. Она представляет собой максимальную энтропию r первых градаций «загрубленной» шкалы, вычисленных при условии, что ξ градаций исходной шкалы объединяются в r первых градаций «загрубленной» шкалы:

$$\Lambda_r(\xi) = \max_{u_1,..,u_r} \left(-\sum_{j=1}^{r} \left(\sum_{k=\sum_{l=1}^{j-1} u_l + 1}^{\sum_{l=1}^{j} u_l} p_k \right) \log_2 \left(\sum_{k=\sum_{l=1}^{j-1} u_l + 1}^{\sum_{l=1}^{j} u_l} p_k \right) \right),$$

$$\sum_{j=1}^{r} u_j \le \xi,$$

$r \le \xi$, ξ=1,2,…,m.

Для вычисления функции Беллмана можно ввести рекуррентное соотношение:

$$\Lambda_1(\xi) = \max_{1 \le u_1 \le \xi} \{-(\sum_{k=1}^{u_1} p_k) \log_2 (\sum_{k=1}^{u_1} p_k)\}$$

$$\Lambda_2(\xi) = \max_{1 \le u_2 \le \xi} \{-(\sum_{k=\xi-u_2+1}^{\xi} p_k) \log_2 (\sum_{k=\xi-u_2+1}^{\xi} p_k) + \Lambda_1(\xi - u_2)\}$$

…

$$\Lambda_r(\xi) = \max_{1 \le u_r \le \xi} \{-(\sum_{k=\xi-u_r+1}^{\xi} p_k) \log_2 (\sum_{k=\xi-u_r+1}^{\xi} p_k) + \Lambda_{r-1}(\xi - u_r)\}$$

Значение функции Беллмана для всей шкалы, т.е. при ξ=m и r=m^* даст, оптимальное значение целевой функции:

$$\Lambda_{m^*}(m) = \max_{1 \le u_{m^*} \le m} \{-(\sum_{k=m-u_{m^*}+1}^{m} p_k) \log_2 (\sum_{k=m-u_{m^*}+1}^{m} p_k) + \Lambda_{m-1}(m - u_{m^*})\} =$$

$$= \max_{u_r} (-\sum_{r=1}^{m^*} (\sum_{k=\sum_{l=1}^{r-1} u_l + 1}^{\sum_{l=1}^{r} u_l} p_k) \log_2 (\sum_{k=\sum_{l=1}^{r-1} u_l + 1}^{\sum_{l=1}^{r} u_l} p_k))$$

В работе проведен расчет выигрыша от оптимального разбиения, на примере задачи оценки прикладного эффекта космического эксперимента (КЭ) планируемого для проведения на российском сегменте МКС. Прикладной эффект КЭ – востребованность (хотя бы потенциальная) эксперимента министерствами, ведомствами, частными организациями и т.п. Для определения p_k использовались данные по оценкам прикладного эффекта по 93 космическим экспериментам.

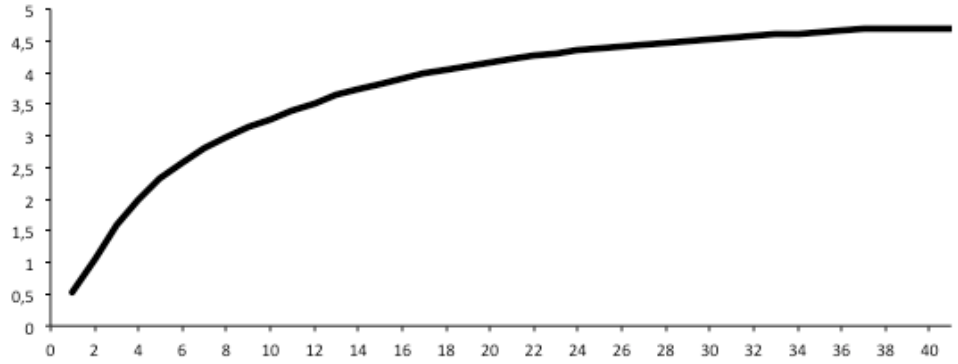


Рис. 3.6. Зависимость энтропии I^* от числа градаций m^*.

На рисунке 3.6. показана зависимость энтропии I^* от числа градаций m^*. По нему видно, что в данном примере не целесообразно использовать больше 36 градаций, так как энтропия

перестает существенно возрастать. Однако при большом числе градаций строить функцию предпочтений затруднительно, т.к. требуется задать предпочтения для большого количества областей. Разбиение на 16 градаций позволяет достичь 83% от максимума, что является хорошим компромиссом между информационными потерями и сокращением шкалы в 16 раз от исходной. Для 16 градаций информационные потери при равномерном разбиении шкалы критерия составляют 28 %, в то время как потери в оптимизированной шкале составили 17 % .

Далее проведен расчет выигрыша от оптимального разбиения, в задаче оценки прикладного эффекта космического эксперимента (КЭ) планируемого для проведения на российском сегменте МКС. Исходные данные и таблица условно-оптимальных управлений приведены в приложении 2. Прикладной эффект определяется на основе 8 показателей:

- эффект в целях безопасности;
- эффект для космической деятельности;
- доступность информации по КЭ для потенциального заказчика;
- КЭ соответствует приоритетным направлениям ФКП (или других целевых программ);
- наличие заинтересованного заказчика;
- социальный эффект;
- экологический эффект;
- экономический эффект.

Каждый из этих показателей оценивается по трехбалльной шкале. Шкала предпочтений прикладного эффекта от КЭ изначально составляла 256 градаций. Необходимо было получить «загрубленную» шкалу из 16 градаций. Для определения p_k использовались данные по оценкам прикладного эффекта по 93 космическим экспериментам. На рисунке 3.7. показана гистограмма плотности вероятности в исходной шкале.

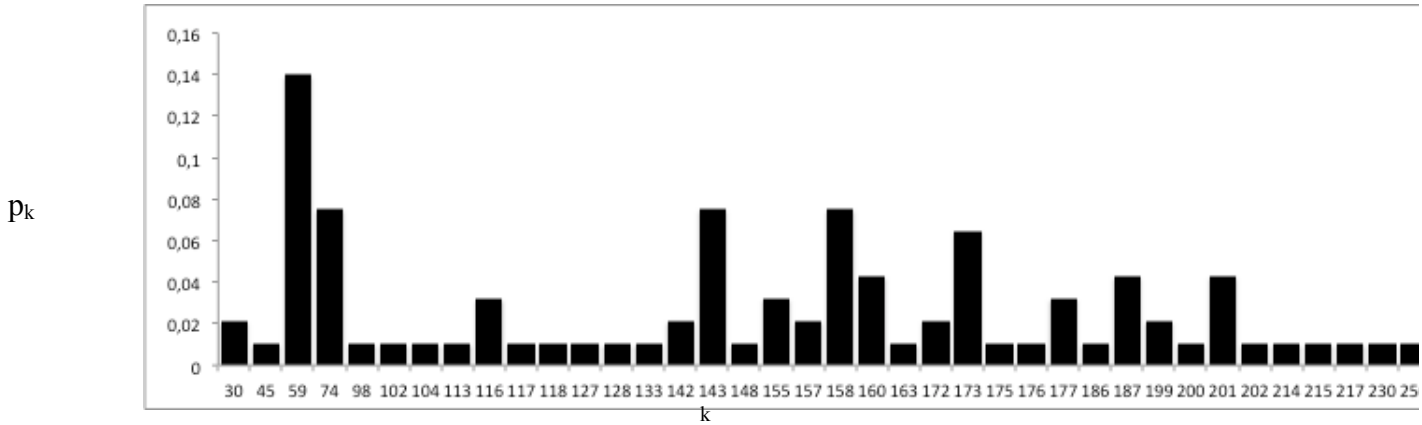


Рис. 3.7. Гистограмма исходной шкалы критерия «Прикладной эффект КЭ».

Для случая равномерного «загрубления» (u_r =16) энтропия составила 3,3991. Значения u_r, полученные в результате применения процедуры оптимизации, приведены в таблице 3.1. Энтропия шкалы «загрубленной» шкалы составила 3,9058.

Табл. 3.1.

										0	1	2	3	4	5	6
r	5	4	5	9		4		4		4			0	3		4

На рисунке 3.8. показана гистограмма полученная в результате применения процедуры дискретизации. Информационные потери при равномерном разбиении шкалы загрубляемого критерия составляют 28 %, в то время как потери в оптимизированной шкале составили 17 %.

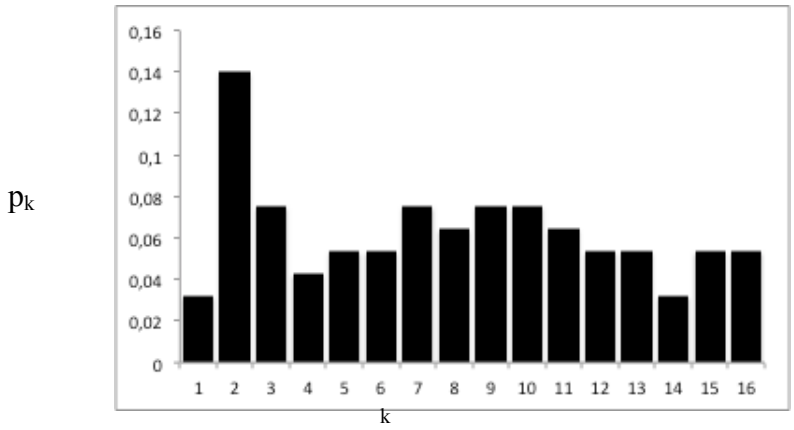


Рис. 3.8. Гистограмма «загрубленной» шкалы критерия «Прикладной эффект КЭ».

3.4. Программная реализация метода

В интерфейсе, показанном на рисунке 3.9., пользователь указывает уровни предпочтений в нижней части экрана. Для каждого уровня предпочтений в верхней части экрана указываются допустимые диапазоны значений критериев. Все заданные уровни предпочтений также видны в списке в нижней части экрана. Так, для текущего уровня предпочтений «Отлично» допустимые интервалы значений критериев выделены звездочкой: Критерий1 - от 0 до 1, Критерий2 - Высокий или Низкий, Критерий3 – 1. Для того чтобы задать невыпуклую область, в данном интерфейсе можно ввести в нижний список 2 и более строк с одним уровнем предпочтений. Таким образом, каждая строка задает один прямоугольный участок для заданного уровня предпочтений.

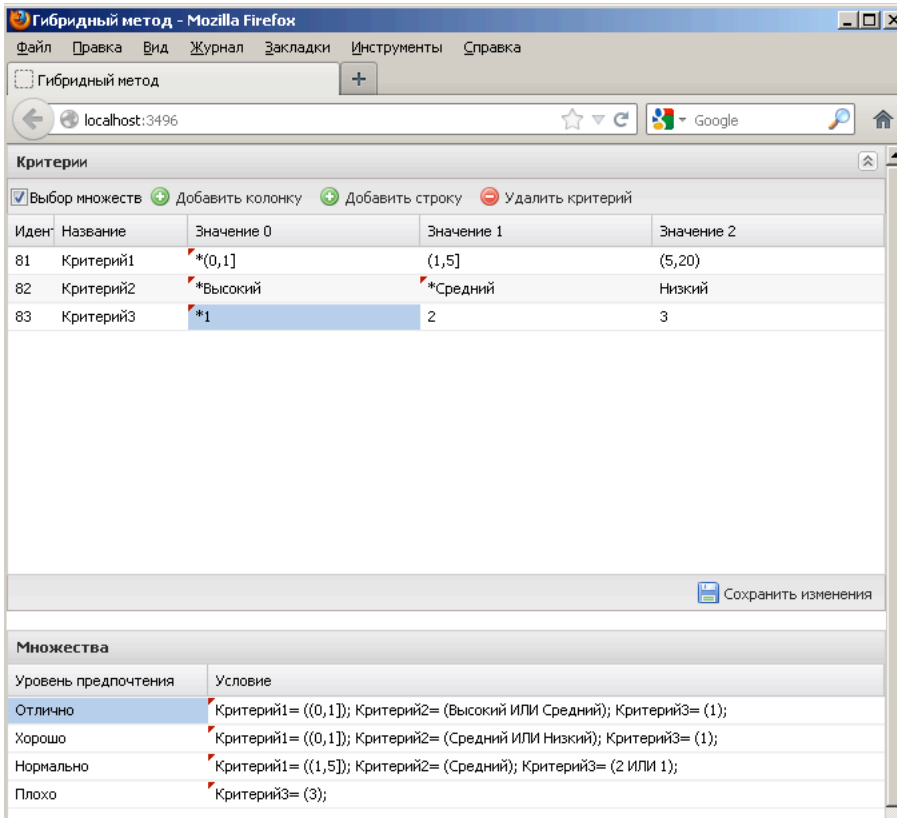


Рис. 3.9. Интерфейса ввода уровней предпочтений для областей.

Приложение разработано с помощью технологии ASP.NET для построения самого приложения, ADO.NET для доступа к базе данных и EXT.NET для отображения данных на стороне клиента. Для хранения данных используется база данных под управлением MS SQL Server.

При загрузке данных в таблицу с критериями из-за неоднозначного количества столбцов используется CodeDOM-граф объектов [150]. Этот объектный граф воспроизводится в виде исходного кода при помощи генератора кода CodeDOM для C#, компилируется в двоичный файл и подключается к программе. После чего, средствами рефлексии адаптер заполняется данными из БД и привязывается к таблице. Остальные таблицы заполняются напрямую из базы данных, т. к. в них фиксированное число столбцов.

Пользователю предлагается выбрать некоторые множества и указать их уровень предпочтения, также добавить критерии, альтернативы, изменить их значения. После ввода данных проверяется, показывают ли множества пользователя какие-либо точки, если да, то такие точки отмечаются, и им ставится в соответствие уровень предпочтения. После этого строится матрица смежности, показывая связи между точками, граф проверяется на наличие циклов с помощью нерекурсивного поиска в глубину. Если циклов не найдено, вычеркиваются строки и столбцы, соответствующие недоминируемым точкам, таким точкам устанавливается результирующий уровень предпочтения. Процесс продолжается, пока матрица не станет пустой. Если же циклы есть, пользователю выводится информация о том, какие из множеств

привели к такой ситуации. Затем все результаты сохраняются в отдельную таблицу в БД. Если необходимо, эти данные выводятся в таблицу с альтернативами.

3.5. Проверка работоспособности гибридного метода построения функции предпочтений

Проверка работоспособности гибридного метода проводилась на примере задачи многокритериальной оценки значимости космических экспериментов (КЭ) [8, 51, 100-107]. В Федеральном государственном унитарном предприятии «Центральный научно-исследовательский институт машиностроения» (ФГУП ЦНИИмаш) значимость КЭ оценивается по следующим критериям:

- научный эффект КЭ;
- прикладной эффект КЭ;
- международное участие;
- технологическая инновационность;
- актуальность КЭ;
- область применения результатов КЭ.

Критерий «Научный эффект КЭ» – это оценка новых фундаментальных знаний, которые рассчитывают получить в результате КЭ. Научный эффект КЭ оценивается в пятибалльной шкале:

0 - Оценка невозможна.

1 - Предлагается повторение ранее проведённых исследований, накопление статистики.

2 - Предложен новый метод для подтверждения (или опровержения) результатов, полученных ранее другим способом.

3 - Предложен решающий опыт (experimentum crucis), который позволит существенно уменьшить имеющиеся неопределённости.

4 - Эксперимент по исследованию ранее никогда не изучавшегося явления, который позволит открыть принципиально новые факты, закономерности (научное открытие), дать старт новому научному направлению.

Для проведения качественных сравнений каждой оценке научного эффекта даны краткие качественные лексические определения 0 – н/у, 1 – неуд., 2- уд., 3 – хор., 4 – отл.

Критерий «Прикладной эффект КЭ» – это востребованность (хотя бы потенциальная) эксперимента министерствами, ведомствами, частными организациями и т.п. Прикладной эффект КЭ оценивался в целочисленной шкале от 0 до 256. Направление улучшения – на максимум. Для использования этой шкалы необходимо разбить ее на интервалы. Для уменьшения информационных потерь разбиение на интервалы проведено с целью

максимизации энтропии оцениваемого критерия. В качестве оценок вероятностей были взяты частоты появления каждого из значений в 93 КЭ. Гистограмма значений Прикладной эффект КЭ в исходной шкале приведена на рис. 3.10.

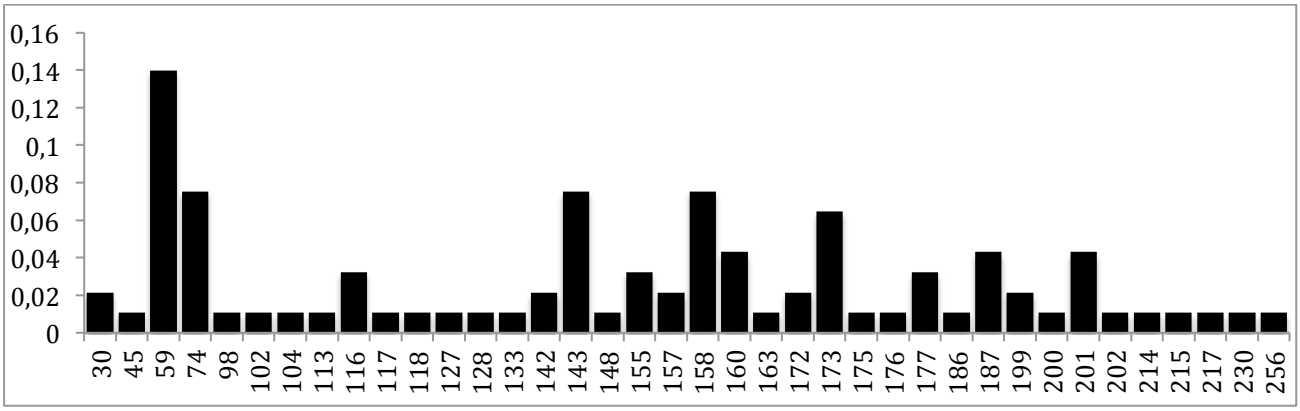


Рис. 3.10. Гистограмма исходной шкалы критерия «Прикладной эффект КЭ».

Если проводить равномерное разбиение исходной шкалы на 4 интервала, то энтропия составит 1,7946. В результате применения предложенной в диссертации процедуры дискретизации шкалы с целью минимизации информационных потерь, энтропия составила 1,9991. Полученная «загрубленная» шкала показана на рис. 3.11. Она разбивает исходное множество значений критерия «Прикладной эффект КЭ» на следующие 4 интервала:

- [0, 98];
- (98, 155];
- (155, 173];
- (173, 256].

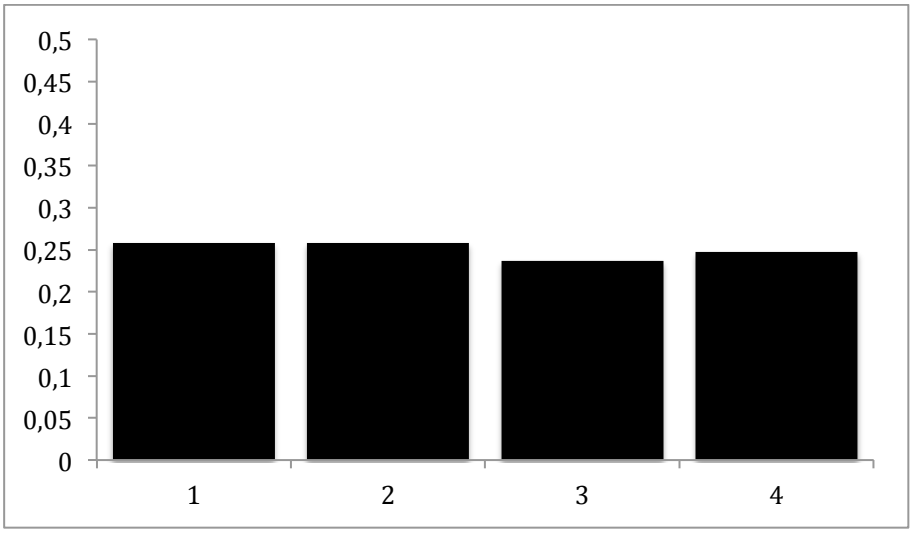


Рис.3.11. Гистограмма «загрубленной» шкалы критерия «Прикладной эффект КЭ».

Критерий «Международное участие» - показывает, является ли КЭ совместным с другими странами. Шкала критерия состоит из 3-х градаций:

0 – не установлено (н/у),

1 – нет,

2 – да.

Критерий «Технологическая инновационность» - показывает долю использования новой техники и технологий. Измеряется в непрерывной шкале в интервале от 0 до 1. На момент проведения проверки работоспособности метода большинство КЭ не были оценены по этому критерию, поэтому значения были заполнены случайными числами, равномерно распределёнными в интервале от 0 до 1, поэтому применение процедуры максимизации энтропии не приведёт к выигрышу. Шкала критерия была разбита на три равных интервала:

- [0, 0,2];
- (0,2, 0,4];
- (0,4, 1].

Критерий «Актуальность КЭ» определяется следующими возможными значениями:

0 - Оценка актуальности КЭ невозможна (кратное наименование – н/у).

1 - Эксперимент в принципе полезен, однако его задержка не повлияет на темп исследований в данной области (кратное наименование – неуд).

2 - Задержка проведения КЭ затормозит, но не остановит дальнейшие исследования в данной области (кратное наименование – уд).

3 - Задержка проведения КЭ приведет к соответствующей остановке исследования в данной области (кратное наименование – хор).

4 - Срыв проведения эксперимента в заданные сроки приведет к принципиальной невозможности в течение многих лет получить требуемую информацию (кратное наименование – отл).

Критерий «Область применения результатов КЭ» задан в четырёх балльной шкале:

0 - Для конкретного подразделения, предприятия (неуд).

1 - Для конкретной отрасли, научного направления в России (уд).

2 - Для народного хозяйства страны в целом (хор).

3 - Мировой уровень (отл).

Таким образом, мощность полного множества альтернатив составляет:

Q = 5 * 4 * 3 * 3 * 5 * 4 = 3600.

Проведем анализ трассировочной печати. Сама трассировка дана в следующем разделе.

Исходные данные по задаче были заданы в формате JSON (строки 3-131). Это достаточно современный формат структурированного представления данных в текстовой строке. Он активно используется в веб-приложениях и при создании интеграционных стыков между различными программными системами. На верхнем уровне задан хэш с четырьмя ключами:

- criterions - описание пространства критериев;

- pref_scale – качественная шкала предпочтений;
- preference – определение областей Mk;
- alternative – названия альтернатив и значения критериев для них.

Поле rank критерия определяет его вес. Для оценки альтернатив, которые попали в одну область или в одну ячейку используется метод взвешенной суммы.

Значения всех критериев были получены на основании суждений экспертов ФГУП ЦНИИмаш. Исключение – «Технологическая инновационность», которая была заполнена по равномерному датчику случайных чисел в интервале от 0 до 1.

Строки с 14 по 17 задают область в критериальном пространстве, где лежат неудовлетворительные значения критериев. Так, ЛПР не устраивает, если хотя-бы по одному из критериев получены неудолетворительные оценки или оценки нельзя определить. Для некоторых областей не указаны значения по некоторым критериям, в этом случае считается, что критерий может принимать любые значения (диапазон значений включает всю область определения). Например, при высоком прикладном эффекте КЭ и отличной области применения результатов, значимость КЭ находится на отличном уровне независимо от международного участия в экспериментах (строки 30-31). Пользовательский интерфейс ввода и редактирования предпочтений пользователя показан на рисунке 3.12. Пользователь указывает уровни предпочтений в нижней части экрана. Для каждого уровня предпочтений в верхней части экрана указываются допустимые диапазоны значений критериев. Все заданные диапазоны значений критериев для каждого уровня предпочтений указаны в списке в нижней части экрана в колонке «Условие». Так для текущего уровня предпочтений «Отлично», допустимые значения критериев выделены звездочкой в верхней таблице. Для того, чтобы задать невыпуклую область, в данном интерфейсе можно ввести в нижний список две и более строк с одним уровнем предпочтений. Таким образом, каждая строка нижнего списка задает один прямоугольный участок для заданного уровня предпочтений.

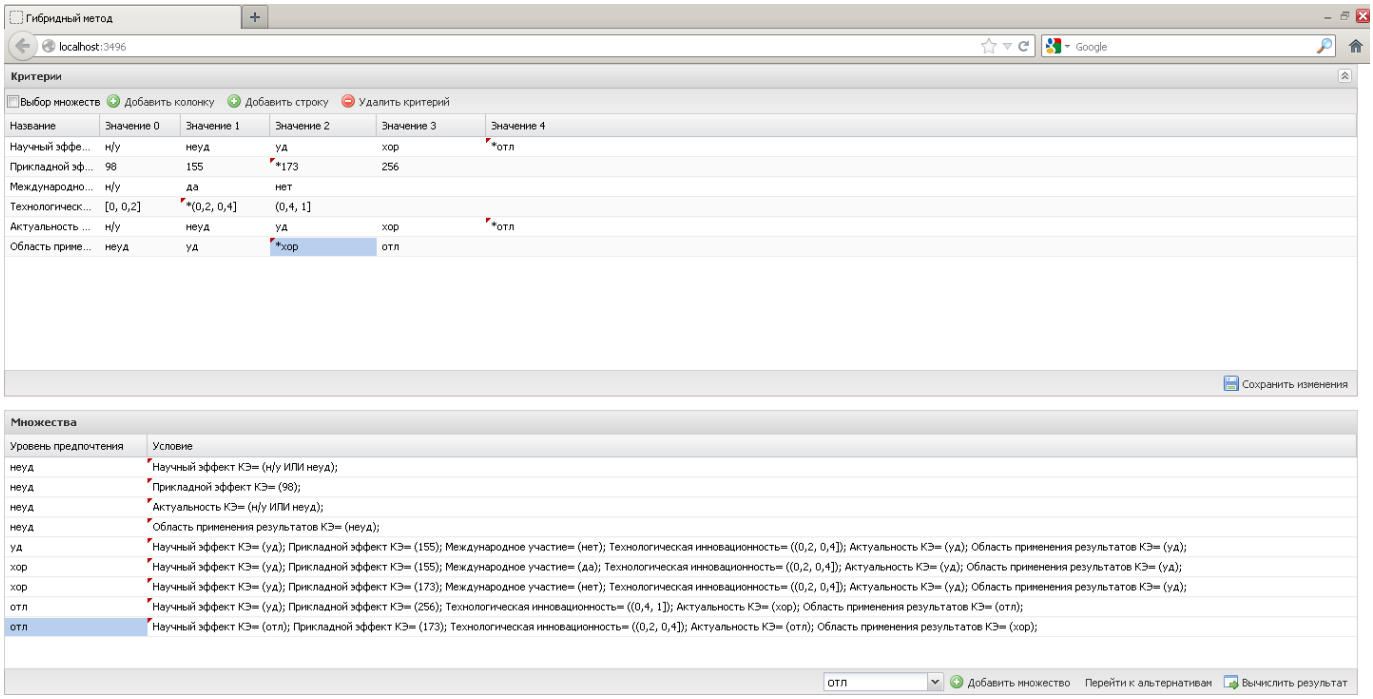


Рис. 3.12. Интерфейс ввода и редактирования предпочтений.

После ввода предпочтений происходит считывание множеств Mk и определение нет ли пересечений между ними (строки 134-159 трассировки).

Далее происходит поиск ячеек в критериальном пространстве, которые не покрыты Mk (в строках 165-266 показаны первые 100 найденных ячеек). Таких ячеек найдено 971. Что существенно меньше полного множества альтернатив (3600).

В строках с 270 по 470 показаны первые 100 отношений доминирования по Паретто и первые 100 отношений доминирования по Подиновскому. Далее массив отношений доминирования дополнен отношениями доминирования, которые указал пользователь (строки 472-494). Всего найдено отношений доминирования: по Парето - 76181, по Подиновскому - 567, пользовательских – 23.

Далее произведен анализ графа на наличие циклов. В строках 500-602 показаны первые 100 путей в графе, которые были проанализированы. Циклов найдено не было. Для того, чтобы убедится, что алгоритм поиска циклов работает корректно, в отдельном разделе приложения 1 «Трассировка при наличии цикла» приведён пример исходных данных в котором найден цикл.

Алгоритм разбора графа показан в строках 605-626. В результате все множество вершин было разбито на 18 уровней.

Далее для всех 93-ти альтернатив было определено, в какую область они попадают и подсчитаны их обобщенные ранги (строки 639-732).

Результаты ранжирования альтернатив показаны в строках 736-828.

3.6. Выводы по третьей главе

1. Предложенный метод построения гибридной функции предпочтений содержит алгоритмы ввода и редактирования системы ценностей ЛПР для высокоразмерного критерия, позволяет учитывать зависимости по предпочтениям.

2. Разработан алгоритм поиска ячеек пространства критериев с целью определения предпочтений на полном множестве альтернатив. Предложен набор правил определения отношения доминирования между областями критериального пространства. Разработан алгоритм анализа графа для определения уровней предпочтений отдельных точек критериального пространства.

3. Гибридный метод является инвариантным по отношению к предметной области, позволяет решать широкий круг задач ракетно-космической отрасли, связанных с многокритриальной оценкой по высокоразмерному критерию, включающему в свой состав как качественные, так и количественны компоненты.

4. Построение многоуровневого дерева агрегирования, в случае большой размерности векторного критерия, приводит к информационным потерям. Разработан метод минимизации информационных потерь при дискретизации шкал обобщенных показателей. Задача рационального разбиения шкалы обобщенного показателя поставлена и решена как задача динамического программирования.

5. Применение метода дискретизации шкал обобщенных показателей в задаче оценки прикладного эффекта космического эксперимента (КЭ), планируемого для поведения на российском сегменте МКС, позволило снизить информационные потери с 28% до 17%.

Глава 4. Создание методологии унифицированной разработки СППР

4.1. Принципы унифицированной разработки

По мнению автора, основная потребность в унификации связана со стремлением сделать максимально широким и доступным весь набор информационных и аналитических ресурсов и сервисов для ЛПР. Современные методы системного и информационного моделирования бизнес процессов (например, технология SADT, язык UML), методы проектирования и системной интеграции информационно-аналитических систем дают возможность не только формализовать с использованием унифицированных подходов основные процедуры выработки и обоснования решений, рекомендуемые теорией, но и предложить решение по программной реализации ресурсов и сервисов СППР, инвариантных по отношению к предметной области [19, 87, 89, 91, 121, 133, 142-144].

Обобщенная схема принятия решений, представленная в работах Э.А. Трахтенгерца, включает следующие этапы:

1. Выявление предпочтений ЛПР в формировании критериев.
2. Использование субъективных предпочтений ЛПР в оценке ситуации (анализе результатов мониторинга).
3. Формирование альтернативных управленческих решений.
4. Оценка альтернатив управленческих решений с учетом предпочтений ЛПР.
5. Учет предпочтений ЛПР в прогнозе результатов принимаемых решений.
6. Выбор лучшего, с точки зрения ЛПР, варианта решения.

Анализ данной схемы принятия решений, позволил сформулировать некоторые принципы, соблюдение которых должно лежать в основе формирования унифицированных требований к программным средствам информационно-аналитической поддержки принятия решений, обеспечивающих приоритет интересов ЛПР.

Унифицированный подход к разработке программного обеспечения СППР состоит из:

- совокупности принципов проектирования;
- каркасного подхода к разработке архитектуры программного обеспечения;
- шаблона правил качественного кодирования;
- набора характеристик оценки качества программного обеспечения СППР.

Унификация заключается в том, что все эти методические рекомендации применимы вне зависимости от того, в какой предметной области необходимо решать задачи поддержки решений. Конечно, некоторые предметные области могут потребовать специфичных проектных

решений, но они не противоречат предложенному унифицированному подходу, а лишь дополняют его.

При проектировании и сборке программного обеспечения СППР используются следующие принципы:

Принцип свободы от субъективизма разработчиков. Система должна быть максимально гибкой и подстраиваться под нужды и потребности ЛПР. При этом настройка решающих правил должна выполняться в интерфейсе системы пользователем, без необходимости изменения исходных текстов программного обеспечения.

Принцип инвариантности по отношению к предметной области. Программное обеспечение СППР должно включать реализацию широкого набора методов работы с векторным критерием, методов оптимизации, содержать инструменты выявления предпочтений пользователя. Программная реализация этих методов достаточно трудоемка, при этом сами методы не привязаны к конкретной предметной области. Поэтому целесообразно разрабатывать унифицированную архитектуру СППР, которая способна работать с произвольной предметной областью.

Принцип множественности методов. Существует множество методов принятия решений. СППР должна обеспечивать легкую и прозрачную интеграцию требуемых методов, советовать пользователю, какой метод в данном случае лучше применим и почему, но не настаивать на нем.

Принцип учета субъективизма лица, принимающего решения (ЛПР). ЛПР наделен полномочиями и несет ответственность за принимаемые решения. Решения, которые предлагает СППР, должны отражать опыт, квалификацию и предпочтения ЛПР. После первоначальной настройки, результаты работы СППР часто не устраивают ЛПР, и нужно предусмотреть процедуру обратного вывода, которая сможет объяснить: какие предпочтения привели к полученному результату и что нужно изменить, чтобы результат работы выглядел иначе.

Принцип дружелюбности к ЛПР. Для работы с СППР не следует требовать от ЛПР специальных знаний теории принятия решений. Интерфейс системы должен наглядно показывать ЛПР все настройки и предпочтения на минимальном количестве экранных форм. СППР должна содержать механизм выявления противоречий в суждениях ЛПР и указывать на них.

Данные принципы в достаточной мере определяют особенности разработки СППР. При создании СППР следует так же руководствоваться общими подходами применяемыми при создании информационно-аналитических систем [1,18,50,81,82, 93, 146, 147].

4.2. Функциональные требования к программному обеспечению СППР

Функциональные требования к СППР охватывают предполагаемое поведение системы, определяя действия, которые система способна выполнять.

Характеристики качественных требований по-разному определены различными источниками. Однако, характеристики, показанные в таблице 4.1., являются общепризнанными [65].

Табл. 4.1.

Характеристика	**Объяснение**
Единичность	Требование описывает одну и только одну вещь.
Завершённость	Требование полностью определено в одном месте и вся необходимая информация присутствует.
Последовательность	Требование не противоречит другим требованиям и полностью соответствует внешней документации.
Атомарность	Требование «атомарно». То есть оно не может быть разбито на ряд более детальных требований без потери завершённости.
Отслеживаемость	Требование полностью или частично соответствует деловым нуждам как заявлено заинтересованными лицами и документировано.
Актуальность	Требование не стало устаревшим с течением времени.
Выполнимость	Требование может быть реализовано в пределах проекта.
Недвусмысленность	Требование кратко определено без обращения к техническому жаргону, акронимам и другим скрытым формулировкам. Оно выражает объективные факты, не субъективные мнения. Возможна одна и только одна интерпретация. Определение не содержит нечётких фраз. Использование отрицательных утверждений и составных утверждений запрещено.
Обязательность	Требование представляет определённую заинтересованным лицом характеристику, отсутствие которой приведёт к неполноценности решения, которая не может быть проигнорирована. Необязательное требование — противоречие самому понятию требования.
Проверяемость	Реализованность требования может быть определена через один из четырёх возможных методов: осмотр, демонстрация, тест или анализ.

Основным пользователем системы является ЛПР. Он может передавать часть своих функций другим пользователям, но нужно понимать, что в этом случае и на принимаемые решения начинают оказывать влияния эти пользователи. Наряду с ЛПР, следует выделить как отдельную личность владельца проблемы - человека, который, по мнению окружающих, должен ее решать и несет ответственность за принятые решения. Но это далеко не всегда означает, что владелец проблемы является также и ЛПР. Конечно, он может быть таковым, но бывают ситуации, когда владелец проблемы является лишь одним из нескольких человек, принимающих участие в ее решении. Он может быть председателем коллективного органа, принимающего решения, вынужденным идти на компромиссы, чтобы достичь согласия. Иногда личности ЛПР и владельца проблемы просто не совпадают. Точно таким же образом некоторые

руководители стремятся переложить на других принятие решений: руководитель полагается на своего заместителя, подписывает подготовленные другими распоряжения. Таким образом, владелец проблемы и ЛПР могут быть как одной, так и разными личностями. С СППР работает именно ЛПР, владелец проблемы работает с СППР, только если он выступает в качестве ЛПР. В качестве ЛПР может выступать и аналитик, но мы не будем термин Аналитик в дальнейшем использовать.

На приведенной на рисунке 4.1. диаграмме вариантов использования СППР показаны основные пользовательские роли, которые существуют в системе. При этом нужно понимать, что один конкретный человек может выступать в разных ролях. Основные функции системы формализованы в виде вариантов использования.

На начальном этапе работы в систему должны быть введены критерии и альтернативы. При этом в формулировании перечня критериев ЛПР должен принимать непосредственное участие. Если забыть про какой-либо важный критерий или задать очень грубую шкалу, то система может выдавать не сбалансированные решения. Техническую работу по вводу этой информации можно поручить оператору по вводу данных. К любой работе ЛПР может привлекать экспертов, но в данном случае ответственность за окончательный перечень критериев и весов несет ЛПР. В терминале системы – привлечение экспертов к вводу критериев и весов – это предоставление им временной роли ЛПР и соответствующих этой роли прав доступа.

Самым ответственным шагом для ЛПР является выбор метода свертки векторного критерия и ввод системы ценностей в пространстве показателей. Ввод системы ценностей организован по разному в зависимости от выбранного метода – это могут быть веса, парные сравнения, идеальные значения или предпочтения, вводимые по сечениям.

Оценку альтернатив по множеству критериев вводят эксперты. В качестве таких экспертов могут выступать не только приглашенные лица, но и сам ЛПР. Если ЛПР дать в системе роль эксперта, то он может вводить оценки по критериям, в данном случае речь может идти о специалистах заказчика СППР, которые могут выступить в двух ролях - ЛПР и экспертов по некоторым критериям.

За организацию прав доступа отвечает администратор системы. Он, в частности, ограничивает права на доступ экспертам только к оценке альтернатив или их части по определенному множеству критериев.

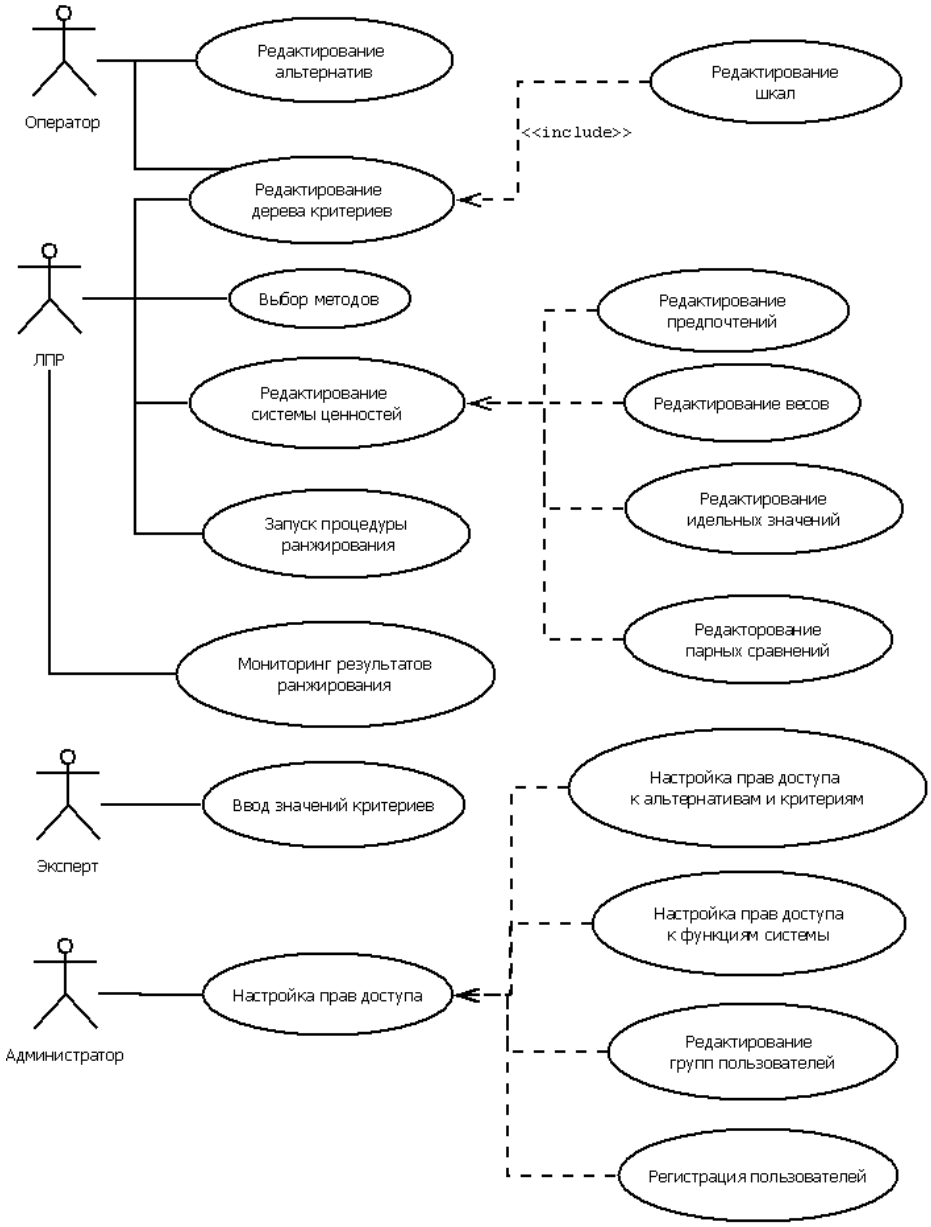


Рисунок 4.1. Диаграмма вариантов использования СППР.

Опишем функциональные требования к системе отдельно для каждой из пользовательских ролей [6-9], представленных на рисунке 4.1.

Функции Оператора СППР:

- регистрация в системе нового оператора;
- вход в систему по имени и паролю;
- ввод, редактирование и удаление альтернатив;
- ввод, редактирование и удаление работ привязанных к альтернативе;
- задание порядка работ;
- ввод, редактирование и удаление критериев;
- задание иерархии критериев и порядка следования критериев;
- ввод, редактирование и удаление шкал;

- ввод направления или улучшений, чем больше, тем лучше или чем меньше, тем лучше, или идеальное значение;
- ввод, редактирование и удаление справочника ресурсов;
- ввод и редактирования потребления ресурсов по каждой работе, привязанной к альтернативе;
- все перечисленные функции должны осуществляться в рамках текущей задачи, на которую есть права доступа;
- указание текущей задачи.

Функции ЛПР:

- регистрация в системе нового ЛПР;
- вход в систему по имени и паролю;
- ввод, редактирование и удаление критериев;
- задание иерархии критериев;
- ввод, редактирование и удаление шкал и рангов лексических значений в шкалах;
- ввод направления или улучшений, чем больше, тем лучше или чем меньше, тем лучше, или идеальное значение;
- выбор метода поддержки решений;
- ввод и редактирование предпочтений;
- ранжирование альтернатив;
- редактирование списка планов;
- запуск процедуры планирования и мониторинг результатов;
- редактирование диаграммы Ганта.

Функции Эксперта:

- регистрация в системе нового эксперта;
- вход в систему по имени и паролю;
- ввод и редактирование значений критериев (непосредственный ввод числовых оценок или выбор из списка лексических оценок).

Функции Администратора:

- ввод, редактирование и удаление задач (вся информация, с которой должна работать система, разделена на отдельные независимые задачи);
- привязка к задаче альтернатив;
- редактирование прав доступа для всех пользователей к тем или иным ресурсам системы;
- ввод, редактирование и удаление ролей;

- назначение разрешений ролям;
- назначение ролей пользователям;
- права доступа должны быть двух видов: права, относящиеся к использованию интерфейсов пользователя и права, относящиеся к работе с объектами системы;
- параметры подключения к базе данных должны задаваться в одном текстовом настроечном файле.

Права, относящиеся к использованию интерфейсов пользователя, назначаются при помощи ролей. К таким правам относятся:

- просмотр и изменение задач;
- просмотр и изменение альтернатив;
- просмотр и изменение критериев;
- просмотр и изменение ресурсов;
- просмотр экспертных оценок и проведение экспертизы;
- просмотр результатов ранжирования и проведение ранжирования;
- просмотр и изменение планов;
- просмотр и изменение данных пользователей;
- просмотр и назначение задач пользователям;
- просмотр и назначение альтернатив пользователям;
- просмотр и назначение ролей пользователям;
- просмотр и изменение ролей;
- просмотр и назначение разрешений ролям.

Права, относящиеся к работе с объектами системы должны включать:

- доступ пользователя к задачам;
- доступ пользователя к альтернативам выбранной задачи для просмотра, редактирования параметров или проведения экспертной оценки.

Система должна быть открыта по отношению к добавлению новых методов поддержки принятия решений. Новый метод должен добавляться без переборки исходных текстов.

4.3. Унифицированная архитектура СППР

Архитектура СППР, реализующая вышеуказанные принципы и требования, строится на основе каркасного подхода, который заключается в создании

- каркаса СППР, отвечающего за базовое функционирование СППР (в том числе, описание пространства критериев и параметров модели предметной области, контроль доступа и информационный обмен),

- множества вариативных модулей СППР (в том числе, формирование альтернатив, ввод и редактирование предпочтений ЛПР разными способами, описание модели предметной области и системы ограничений, многокритериальная оценка и оптимизация решений различными методами), которые подключаются через так называемые точки расширения каркаса СППР (точка расширения описывается как спецификация интерфейса взаимодействия каркаса с множеством модулей, которые к нему подключаются).

Каркас СППР обладает следующими функциями:

- реализует механизм описания пространства критериев и параметров модели предметной области в единой информационной среде;
- позволяет пользователю выбирать методы поддержки решений и организует информационный обмен между ними;
- обеспечивает хранение, отображение и редактирование справочника атрибутов альтернатив;
- обеспечивает контроль доступа.

Контроль доступа должен быть организован как к отдельным функциям СППР, так и к отдельным критериям, и к отдельным альтернативам.

Остальные функции СППР реализуются через следующие точки расширения (гнёзда) каркаса:

- ввод и редактирование предпочтений ЛПР. Возможно применение различных методов теории принятия решений для ввода системы ценностей ЛПР, поэтому это поведение выделено в вариативное;
- формирование альтернатив;
- описание системы ограничений для параметров модели предметной области, с целью определения множества допустимых решений;
- описание модели предметной области, в том числе связей между параметрами модели предметной области и критериями эффективности найденных решений;
- многокритериальная оценка или кластеризация альтернатив с учетом введенных предпочтений;
- поиск допустимых решений выбранным методом оптимизации;
- экспорт и импорт данных.

При настройке соответствующего модуля в точке расширения указывается, для какого подмножества альтернатив и критериев он будет использоваться. На рисунке 4.2. приведена схема унифицированной архитектуры СППР, созданной на основе данных принципов.

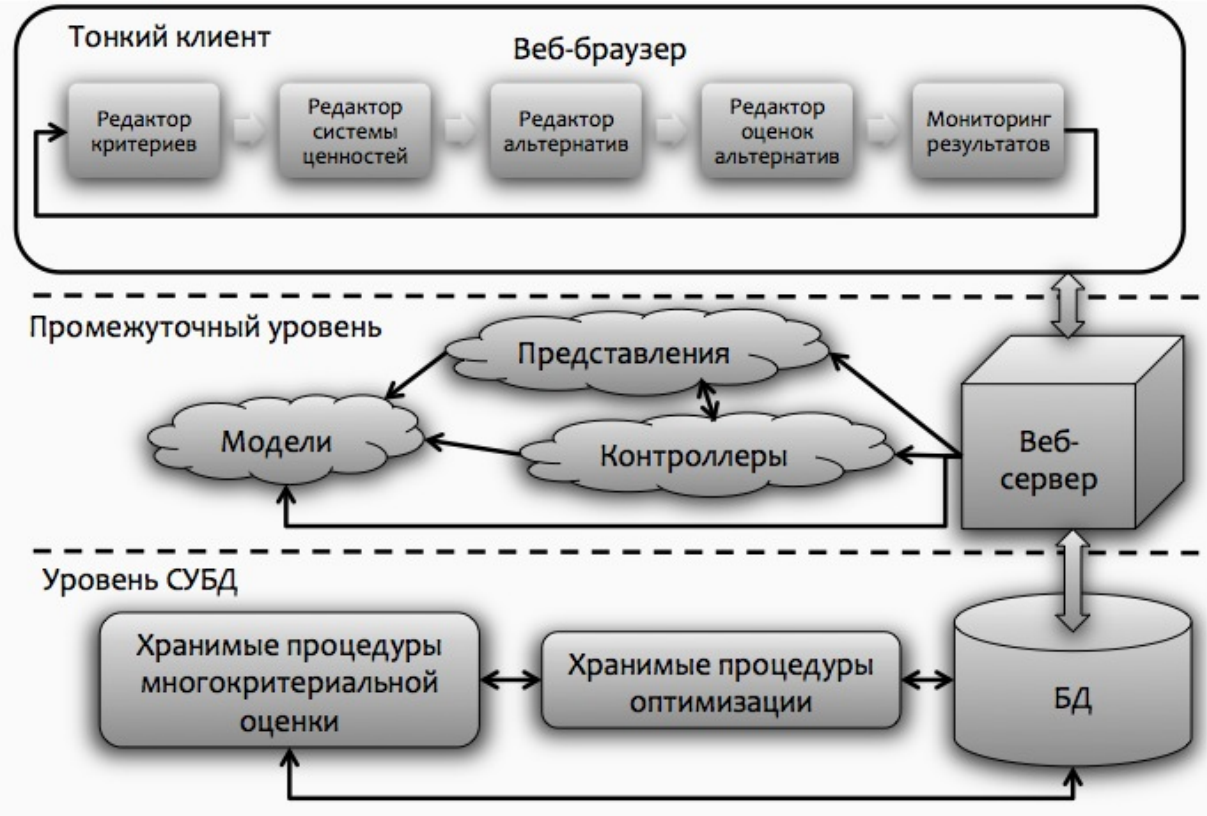


Рис. 4.2. Схема унифицированной архитектуры СППР.

СППР целесообразно реализовывать в трехуровневой архитектуре:

Нижний уровень – это ПО под управлением СУБД, отвечающее за хранение всей информации в единой базе данных, реализующее алгоритмы многокритериальной оценки альтернатив и хранимые процедуры оптимизации модели предметной области. На этом уровне допускается динамическое подключение новых хранимых процедур для новых методов многокритериальной оценки, для новых моделей предметных областей и новых методов оптимизации.

Промежуточный уровень – это программное обеспечение, отвечающее за обработку введенной пользователем информации и подготовку необходимых форм ввода-вывода данных. На промежуточном уровне реализуются шаблоны новых форм ввода и редактирования предпочтений ЛПР и шаблоны форм ввода альтернатив.

Верхний уровень (тонкий клиент) – это готовое программное обеспечение браузера, в котором отображаются все веб-формы, подготовленные на промежуточном уровне и происходит отправка введенных пользователями СППР данных. Именно на верхнем уровне с СППР взаимодействует ЛПР.

В результате трехуровневой организации с Web-интерфейсом мы получаем:

- на компьютер пользователя не требуется установка дополнительного ПО;
- пользователь работает в интернет браузере – привычной для него программе и при необходимости может легко менять компьютеры и операционные системы;

- данные хранятся и обрабатываются на серверах, что повышает уровень их защищенности;
- высокая производительность требуется только для сервера и мы можем её увеличивать добавляя новые сервера.

Разработка программного обеспечения должна базироваться на принципах открытости по отношению к новым методам свертки векторного критерия и новым методам оптимизации. Все исходные тексты должны быть помещены в открытом репозитории на базе одной из систем управления версиями: GIT, SVN, Fossil.

4.4. Унифицированная модель информационного обеспечения СППР

Структура базы данных должна разрабатываться так, чтобы систему можно было настраивать на решение задач из различных предметных областей в единой информационной среде. Таким образом, при разработке СППР обеспечивается свойство инвариантности использования по отношению к предметной области. На рисунке 4.3. показан фрагмент унифицированной концептуальной модели реляционной базы данных (БД) СППР описывающий пространство критериев и оцениваемые альтернативы. Указанная модель соответствует третий нормальной форме. Схема составлена согласно стандарту IDEF1X, который учитывает такие требования, как простота изучения и возможность автоматизации[53].

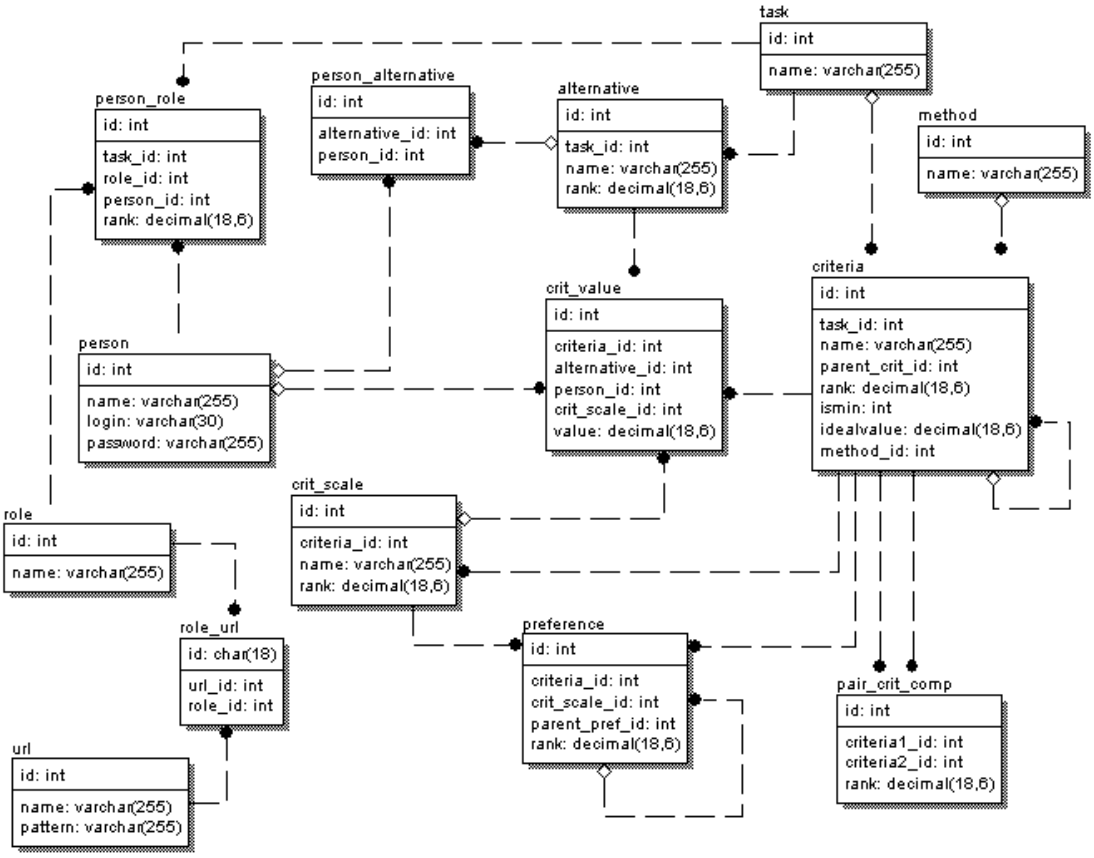


Рис. 4.3. Фрагмент унифицированной концептуальной модели БД СППР.

Сущность task отвечает за представление отдельных задач, решаемых СППР. Каждая задача (task) характеризуется своим множеством альтернатив, которые на рисунке 4.3. представлены сущностью alternative. Задача (task) с векторным критерием, компоненты которого хранятся в сущности criteria. Отдельные критерии могут быть организованы в многоуровневое дерево, для этого используется атрибут prent_crit_id у сущности criteria, который указывает на «родительский» критерий. Доступные методы агрегирования компонент векторного критерия указываются в сущности method.

Основной идеей унифицированной модели БД СППР является обеспечение структурной независимости от конкретных оцениваемых объектов. Для альтернатив, критериев и значений критериев применяется подход аналогичный модели Entity–Attribute–Value(EAV). Для этого в сущности crit_value содержатся значения критериев для всех альтернатив. Преимуществом данного подхода является то, что добавление новых объектов предметной области не требует перестроения модели. В единой сетевой базе данных СППР можно хранить разнородные задачи с различными критериями оценки и деревом их агрегирования с произвольным числом уровней вложенности. Пользователям системы обеспечивается разграничение доступа на уровне ролей (сущность person_role), которые им назначены в конкретных задачах поддержки принятия решений. Кроме того, права доступа разграничиваются на уровне отдельных альтернатив (сущность person_alternative), таким образом, экспертам можно поручать оценку альтернатив только из того подмножества, в котором они наиболее компетентны.

Унифицированная концептуальная модель реляционной БД СППР, показанная на рисунке 4.3., позволяет:

- хранить и обрабатывать в реляционной базе данных векторный критерий произвольной размерности;
- проводить агрегирование компонент векторного критерия и получить в результате иерархию критериев с любым количеством уровней вложенности;
- интегрировать произвольные методы свёртки векторного критерия;
- задавать экспертные или объективные оценки по различным альтернативным вариантам управленческих решений;
- разграничивать единое информационное пространство базы данных СППР по отдельным задачам с индивидуальными уровнями доступа пользователей;
- проводить обработку исходных данных для многокритериального анализа на уровне сервера управления базами данных.

4.5. Обеспечения качества программного обеспечения СППР

При программной реализации СППР крайне важно обеспечить высокий уровень качества создаваемого ПО. Рассмотрим правила качественной разработки СППР. Список этих правил не претендует на полноту. Затруднительно установить первоначальный источник отдельных правил из этого списка, они встречаются и обсуждаются на сайтах www.wikipedia.org, www.habrahabr.ru, в работах Р. Гласса [50]. Автором проделана работа по выделению из всех рекомендаций по качественному кодированию программного обеспечения тех правил, которые целесообразно соблюдать при разработке СППР. Эта работа была проведена на основе личного опыта разработки ряда СППР и систем мониторинга. На базе этого списка правил проектировщик СППР должен разрабатывать свой список, специфичный для выбранных средств разработки и с учетом мнения коллектива программистов. Важно, чтобы среди всех разработчиков СППР было достигнуто согласие относительно этих правил. Ниже приведены основные правила качественной разработки СППР:

- ✓ Все исходные тексты программ, кроме временных скриптов, должны храниться в единой системе контроля версий.
- ✓ Недопустимо дублировать одну и ту же логику в разных частях кода без веских на то оснований. Основаниями могут быть: техническая невозможность избежать дублирования, обусловленная спецификой среды разработки; невозможность иного способа достижения приемлемой производительности.
- ✓ Недопустимо использование в исходном тексте «магических чисел». «Магические числа» - это числовые значения, которые встречаются в исходном тексте программы, кроме чисел 0 и 1 и кроме инициализации констант в начале программы. В программе не должно быть «закомментированного кода». «Закомментированный код» - это часть исходного текста программы, которая перестала выполняться, так как была помещена внутрь комментария /* .. */, (*...*) и т.д.
- ✓ Переменные, структуры данных, таблицы должны объявляться в начале каждого скрипта, объявление их в других местах недопустимо.
- ✓ Все добавления, изменения и удаления в таблицах должны сохраняться в таблице журнала изменений.
- ✓ Все скрипты должны быть в единой кодировке. В качестве единой кодировки рекомендуется выбрать UTF-8.
- ✓ Группу связанных изменений базы данных необходимо выполнять внутри транзакции.

- ✓ Названия любых объектов (переменные, поля, таблицы) должны отражать смысл своих значений.
- ✓ В части FROM запроса, после присоединения новой таблицы оператором JOIN в условии ON, должны встречаться только поля логического первичного ключа присоединённой таблицы, при этом должны встречаться все поля этого первичного ключа.
- ✓ Если есть условия, непосредственно не связанные с правилом объединения связываемых таблиц, а накладывающие ограничения на итоговый результирующий набор данных, они записываются в секции WHERE запроса.
- ✓ Во всех случаях, кроме разовых временных запросов, недопустимо использовать конструкцию SELECT INTO для создания таблиц. Таблица должна быть явно создана в начале скрипта.
- ✓ При объединении двух таблиц всегда необходимо использовать оператор JOIN. В случае, если производится внешнее объединение - LEFT OUTER JOIN.
- ✓ Курсоры и циклы в SQL допустимо использовать, только если задачу нельзя решить иначе.
- ✓ Если в select-е более одной таблицы, то запрещена ссылка на поля без имени или псевдонима имени таблицы.
- ✓ Результат не должен зависеть от неконтролируемого разработчиком порядка выборки данных самим SQL сервером. Запрещено использовать top без order by.
- ✓ Все свойства форматирования веб-страниц должны устанавливаться через CSS файл.
- ✓ Обращение к базе данных с веб-сервера должно происходить только через хранимые процедуры. Таким образом достигается оперативность изменений бизнес-логики, повышается производительность и гибкость, обеспечивается более высокий уровень безопасности.

СППР должна содержать рекомендации по выбору методов теории принятия решений, применяемых при многокритериальном анализе альтернатив. Причем этот выбор зависит от множества условий, которые накладываются конкретной решаемой задачей. Кроме самого метода важно качество его программной реализации. Необходимо учесть пожелания ЛПР по точности решения, по времени, которое он может потратить на ввод предпочтений, по быстродействию при поиске рационального решения и по другим факторам. Все эти факторы могут быть разделены на 2 группы: свойства присущие методу теории принятия решений (обычно это метод скаляризации векторного критерия), который применяется; и свойства присущие программной реализации данного метода.

В первую группу свойств входят:

- точность (оценивается как процент совпадений при выборе альтернатив с использованием данного метода поддержки решений и при непосредственном выборе без использования каких-либо методов в задачах, в которых ЛПР не испытывает затруднений с непосредственным выбором);
- трудоемкость ввода и редактирования предпочтений для ЛПР (оценивается как суммарное время требуемое для ввода весов, парных сравнений и т.д.);
- размерность по числу критериев, числу градаций на шкалах, числу альтернатив (оценивается максимальная размерность с которой ЛПР может работать);
- алгоритмическая сложность (асимптотическая сложность O(g(n,m)) алгоритма ранжирования, отбора альтернатив или оптимизации, где n - число критериев и m – число альтернатив);
- понятность (время требуемое на изучение метода);
- требования к исходным данным по оцениваемым альтернативам (лексические оценки значений критериев, количественные оценки значений критериев, нечеткие оценки значений критериев).

Свойства программной реализации СППР должны оцениваться по существующим стандартам качества ISO 9126. К требуемым свойствам программной реализации СППР относятся:

- удобство интерфейса, простота изучения и простота использования;
- эффективность по времени отклика системы;
- эффективность времени поиска решения (оценка максимального, минимального и среднего времени решения задачи заданной размерности на вычислительных средствах ЛПР, этот показатель может отличаться от алгоритмической сложности в случае неэффективной программной реализации);
- эффективность использования ресурсов (оперативной памяти, процессоров, внешних запоминающих устройств) ;
- переносимость;
- масштабируемость;
- интегрируемость;
- надежность (способность СППР в течение заданного срока работы с заданной вероятностью функционировать без ошибок и сбоев);
- защищенность.

4.6. Формализация процесса разработки СППР

На протяжении последних десятилетий управление разработкой ПО является задачей повышенной сложности. Существуют разные мнения относительно того, насколько жестко должен быть регламентирован процесс разработки программного обеспечения. Некоторые методологии, например XP, придают большее значение неформальному общению участников процесса (программистов, заказчиков, менеджеров и др.). В XP нет необходимости жестко специфицировать требования заказчика и регламенты поведения разработчиков [91]. Другие методологии, например RUP, настаивают на жесткой регламентации рабочих процессов и ролей участников проекта [163]. Есть и промежуточные подходы [89]. Периодически возникают споры о том, какой подход лучше. Предметом раздела является не выбор лучшей из них, хотя это, безусловно, важный вопрос, а разработка инструментария, который позволяет автоматизировать процесс управления разработкой СППР, используя ту методологию, которую предпочитает лицо, принимающее решения (директор по информационным технологиям, начальник отдела разработки ПО, руководитель группы разработчиков).

Обычно крупные компании не реализуют какую-либо методологию в «чистом виде». Они берут одну из них за основу и адаптируют её под конкретные нужды. Большинство авторов считают методологию набором шаблонов, который позволяет эффективно строить собственные процедуры и правила работы. Однако при любой методологии необходимо, чтобы процесс работы был описан, был управляемым и поддавался измерению, т.е. чтобы и сам процесс и его результаты можно было оценить.

Для описания процессов разработки ПО можно использовать обычные текстовые документы на естественном языке. Наряду с ним применяют описания на формальных языках с применением графоаналитической записи: UML, BPEL, Aris и др. Графоаналитические языки обеспечивают следующие преимущества: наглядность и простота записи, однозначность в определении сложных процессов и объектов, лаконичность. Далее будет показано, как использование графических представлений UML позволяет отразить процесс разработки СППР.

Использование UML объясняется тем, что он широко распространен, бурно развивается, понятен читателям разной квалификации и специализации (менеджеры, аналитики, программисты, высшее руководство), и позволяет отразить как динамические, так и структурные аспекты моделируемой системы управления [39]. В работе В.С. Авдуевского, В.П. Осипова и В.П. Шалимова [3] можно найти описание процессов разработки ПО на UML применительно к методологии RUP. Однако RUP весьма сложен и потому он целесообразен в

больших проектах. Предлагаемая нами методология формализации процессов создания СППР достаточно эффективна в средних и малых проектах.

На рисунке 4.4. показана диаграмма Use Case, которая дает представление о ролях участников разработки ПО и их функциях в системе управления жизненным циклом СППР. Определяется не просто что должна делать система, а что должна делать система для конкретного пользователя. Именно ориентация на пользователя отличает подход предложенный создателями Use Case. Кроме того Use Case диаграммы используются для описания системы разделения прав, обязанностей и ответственностей.

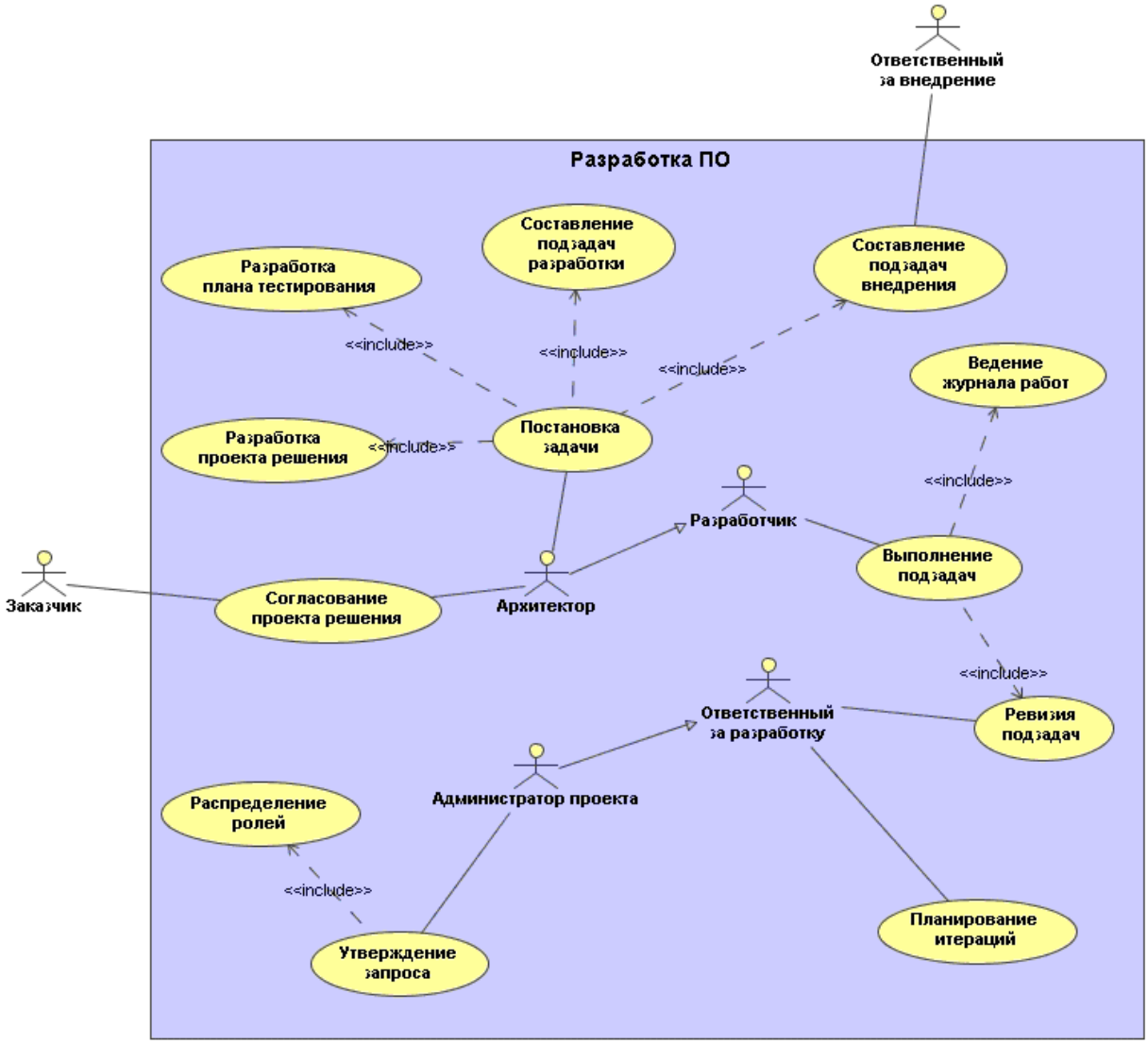


Рис. 4.4. Диаграмма Use Case для управления жизненным циклом СППР.

На рисунке 4.4. показана Use Case диаграмма, которая дает общее представление о функциях системы управления разработкой СППР. Динамика поведения системы будет показана ниже на рисунках 4.5.-4.6. в виде диаграммы деятельности и диаграммы активностей. Часто из диаграммы деятельности можно вывести диаграмму состояний. Иногда на диаграммах деятельности явно указывают объекты и их состояния. Диаграммы деятельности - это аналог

схем алгоритмов, а диаграммы состояний задают конечный автомат в виде графа состояний с указанием условий перехода.

Процесс создания СППР удобно рассматривать как выполнение совокупности запросов на разработку ПО. Состояния запроса на разработку показаны на рисунке 4.5.

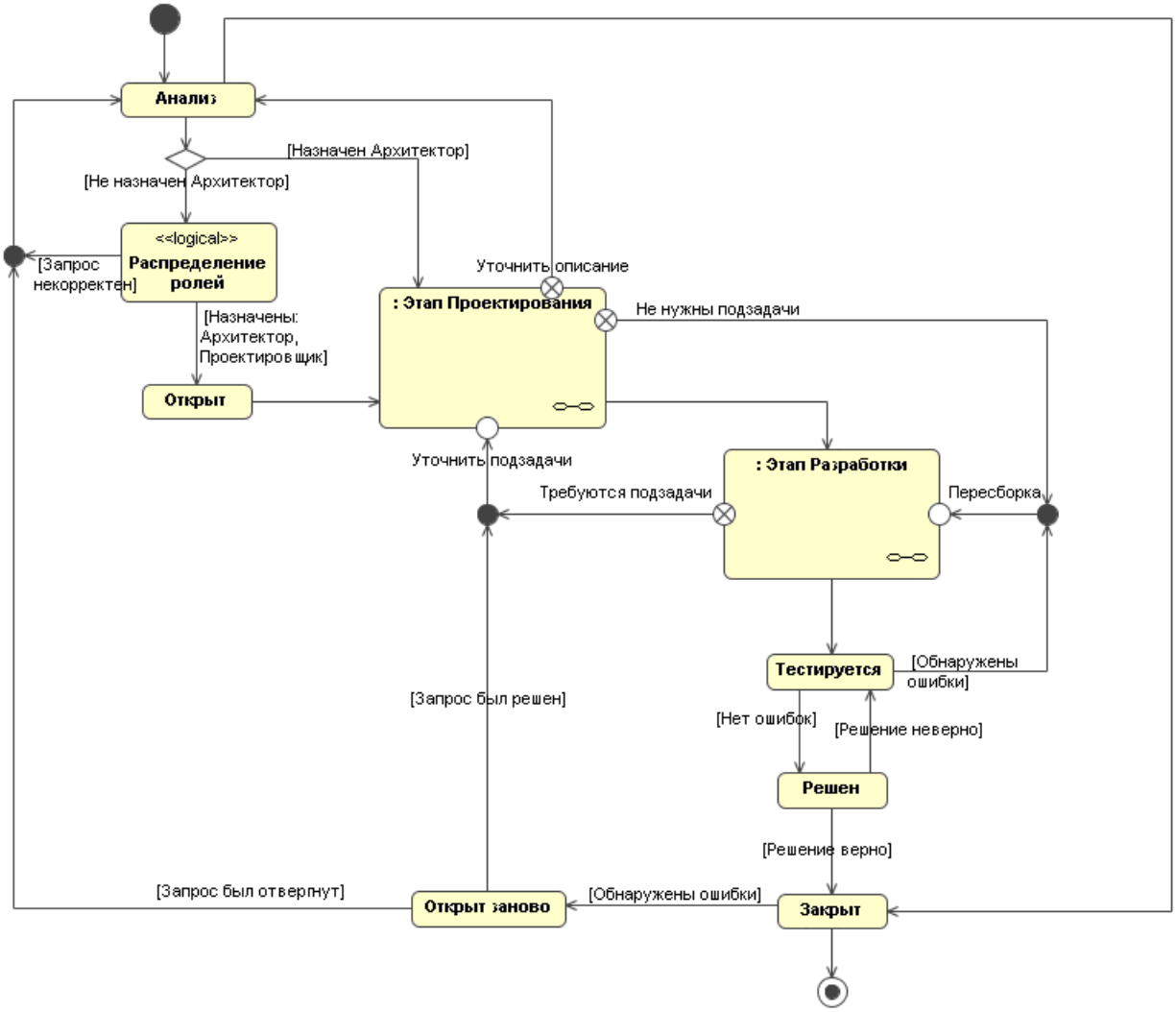


Рис. 4.5. Диаграмма состояний процесса выполнения заявки на разработку ПО.

На диаграммах показаны вложенные композитные состояния. С точки зрения организации самого процесса разработки важно создать взаимосвязи между запросами, т.е. определить, какие задачи необходимо решить, чтобы завершить выполнение запроса. В состав запроса на разработку обычно входит несколько небольших подзадач. В этом случае проще оценить срок их выполнения.

Процесс разработки СППР имеет смысл организовывать итеративно. В процессе разработки возникают риски невыполнения работы в срок, ошибочного понимания требований к СППР, ошибок при реализации требований. Для их снижения итерации должны быть короткими: одна-две недели. Процесс планирования каждой итерации фиксируется на диаграмме деятельности (см. рисунок 4.6.). При планировании исполнители подзадач сами

выбирают, какими подзадачами они хотели бы заниматься, и сами оценивают свои трудозатраты на них.

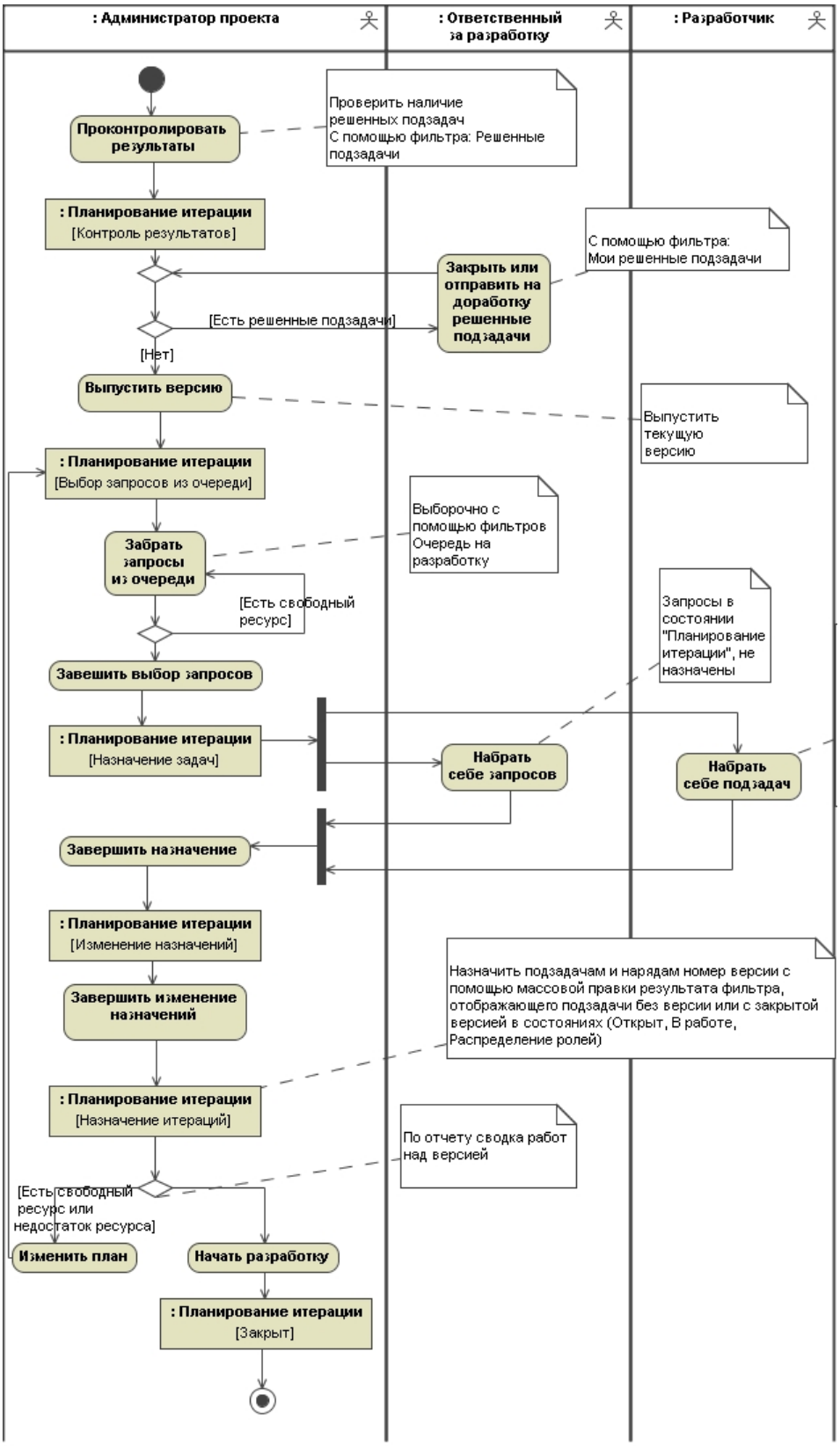


Рис. 4.6. Диаграмма планирования итераций.

Чтобы эффективно осуществлять управление разработчиками ПО, необходимо фиксировать не только их основную деятельность, но и все вспомогательные задачи которые

они решают. Все работы должны быть отражены на соответствующих диаграммах. В частности, на рисунке 4.7. показано, как фиксируется процесс консультаций сотрудников разработчиками ПО.

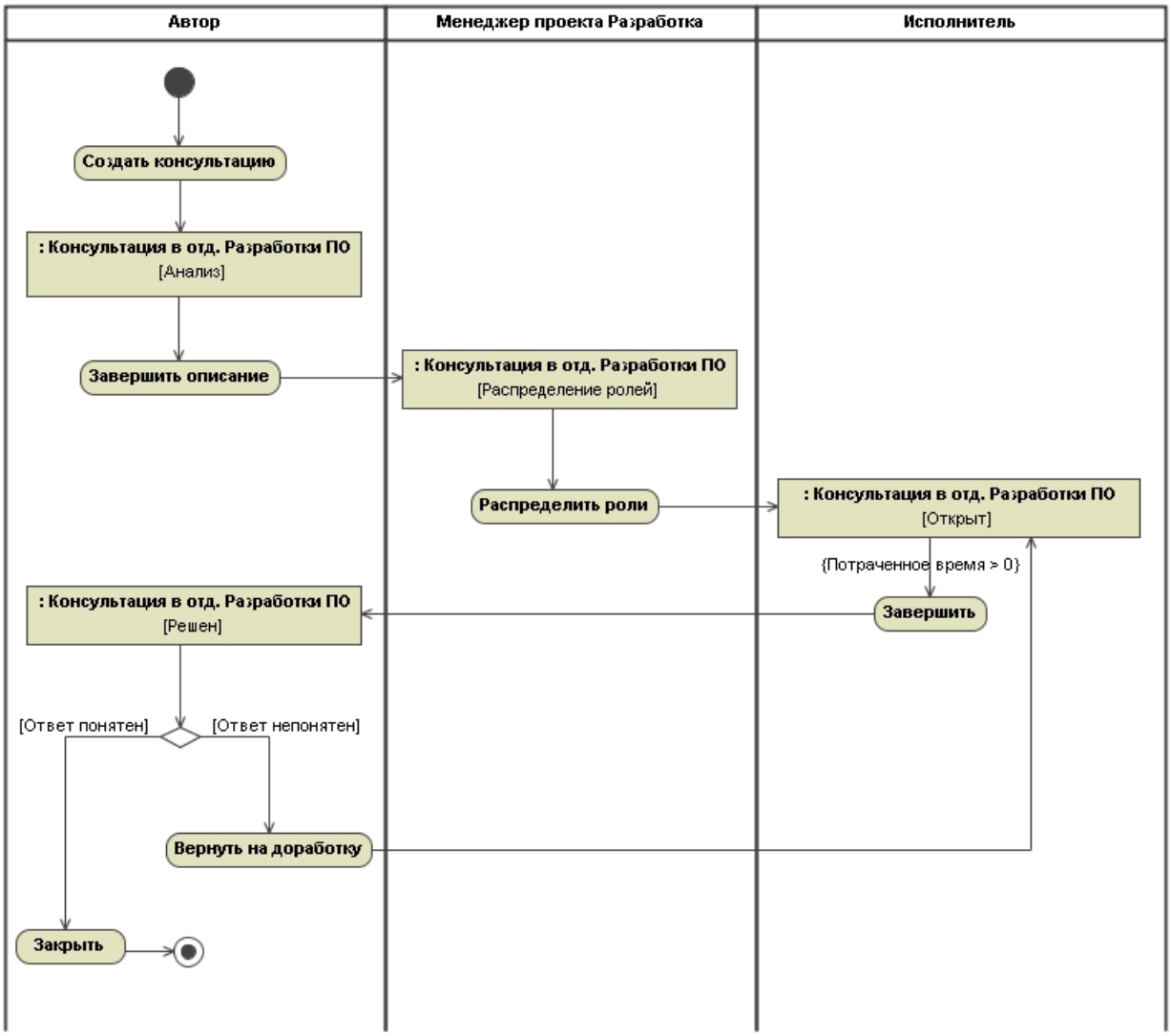


Рис. 4.7. Диаграмма консультаций в отделе разработки ПО.

Сам процесс разработки диаграмм, регламентов и положений (см. ниже) формализуется через систему нарядов. Диаграмма состояний наряда показана на рисунке 4.8., а процесс его исполнения на рисунке 4.9.

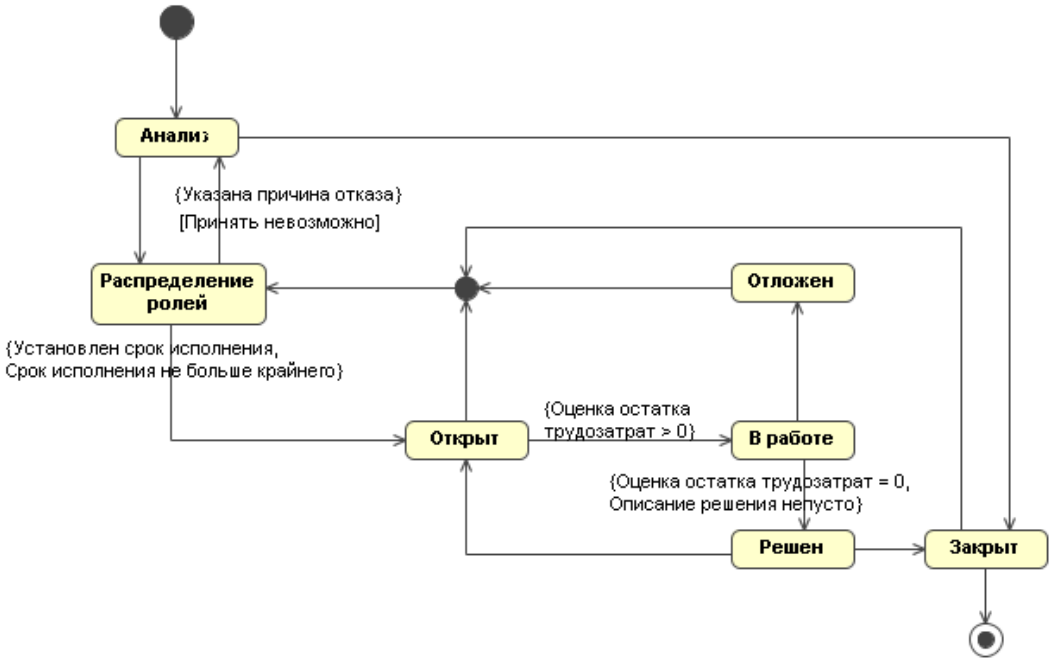


Рис. 4.8. Диаграмма состояний наряда.

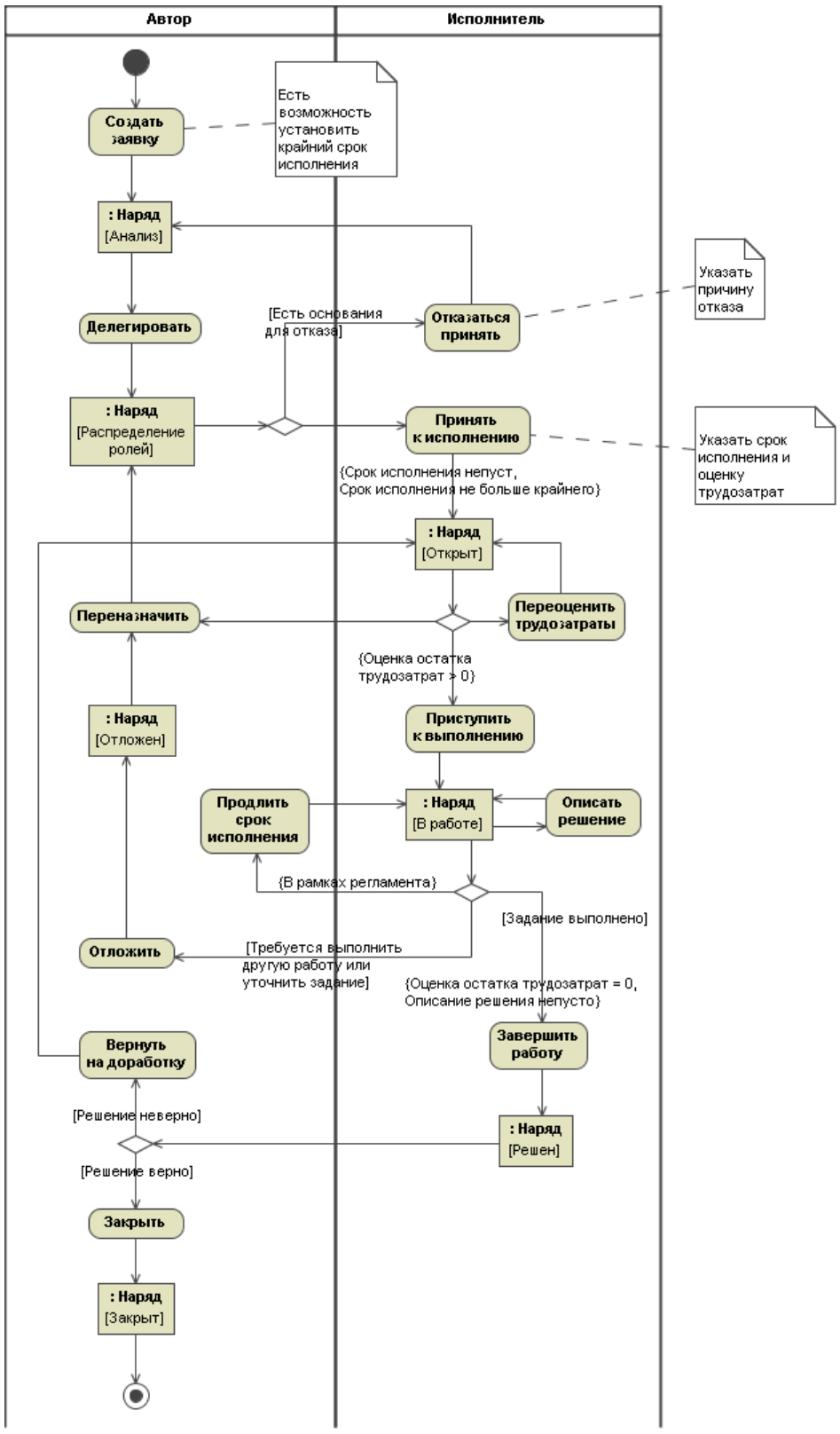


Рис. 4.9. Диаграмма исполнения наряда.

Изложенный выше материал может рассматриваться как формализация процесса разработки программного обеспечения СППР в виде диаграмм UML, на основе которых автоматизируется управление этим процессом.

4.7. Разработка механизма сбора информации о процессах жизненного цикла СППР

Для учета работ на всех стадиях жизненного цикла ПО можно разработать свою систему или воспользоваться существующими для этой цели готовыми программными продуктами.

Первый путь связан с большими расходами, но хорош тем, что специфика конкретных процессов разработки СППР может быть достаточно просто учтена.

Второй путь предполагает использование как платного, так и бесплатного готового ПО. Как правило, платные решения, в конечном счете, часто оказываются выгодней с финансовой точки зрения, чем собственные разработки [34]. Большинство таких продуктов ориентировано на один из этапов жизненного цикла ПО: управление требованиями, управление процессом разработки, управление устранением ошибок и т.д. При использовании разных программных продуктов возникает задача их интеграции, что приводит к возрастанию накладных расходов на сопровождение.

Необходимо, чтобы программные продукты обладали следующими функциональными свойствами:

- позволяют проводить гибкие настройки;
- позволяют включать в свой состав дополнительные модули;
- поставляются с открытыми исходными текстами.

Их можно применять не только на тех этапах, для которых они предназначались изначально. Это поможет экономить на интеграции или на собственной разработке.

В технические требования к системам управления жизненным циклом включают:

- поддержку современных мощных СУБД;
- наличие Web интерфейса пользователя;
- поддержка стандартных протоколов взаимодействия с другими системами;
- поддержка распространенных форматов экспорта/импорта документов.

По первому требованию желательна поддержка той СУБД, которая уже эксплуатируется в организации, т.к. это упрощает сопровождение. Построение OLAP системы обычно реализуется на базе первичной информации, хранящейся в СУБД.

Web интерфейс позволяет упростить взаимодействие с удаленными разработчиками, т.к. он не требует установки специального программного обеспечения на их компьютеры. Web интерфейсы распространяются все шире, т.к. позволяют работать с системой с любого компьютера, подключенного к Интернет; облегчается сопровождение системы. Удобство работы пользователя с Web интерфейсом у сложных систем не хуже, чем у традиционных оконных приложений.

Есть несколько систем из числа программ, задействованных в управлении разработкой СППР, которые удовлетворяют заданным техническим требованиям – это DotProject, HP OpenView, XPlanner, Borland Cariber. Однако настройка на заданные функции в них реализована не достаточно гибко. По нашему мнению, среди широко распространенных систем, наиболее полно и качественно заданным требованиям удовлетворяет система JIRA, разработанная австралийской компанией Atlassian. Ещё более высокий уровень настроек обеспечивает российская система TrackStudio, разработанная ООО «Гран». Однако для этого программного продукта, как и для других новых продуктов, ещё не завоевавших широкого распространения на рынке, существуют риски потери поддержки и прекращения выхода новых версий.

Система JIRA - это средство для управления проектами и запросами. Она может применяться во всех случаях, когда необходимо организовать работу сотрудников, назначать им задания и оперативно контролировать их выполнение. Пользователями JIRA являются все участники разработки ПО.

С её помощью можно организовать контроль разработки проектов, можно определить свой собственный метод движения запросов согласно UML-диаграммам (см. рисунки 4.9.-4.10), сконфигурировать правила уведомления всех участников процесса о событиях, управлять правами доступа пользователей и делать многое другое.

Можно сказать, что JIRA – это пример системы использующей современный подход к разработке программного обеспечения. По сути, она позволяет программировать свое поведение без программирования в традиционном смысле этого слова, т.е. без написания кода на каком-либо языке программирования. Программирование в JIRA заключается в указании множества настроек, создании правил выполнения и включении в состав системы необходимых дополнений. Причем эти действия выполняются в визуальном Web интерфейсе или путем разработки XML файлов со схемами, описывающими структуру системы и алгоритмы её работы. Пользователь сам выбирает, какой способ программирования ему больше подходит. На рисунке 4.10. показана схема классов настроек, используемых при программировании системы. Создание конфигурации, отражающей конкретные процессы исполнения запросов, происходит быстрее, чем при обычном программировании систем. Так, для того, чтобы реализовать процессы выполнения запросов на разработку, приведенные на UML-диаграммах (рисунки 4.5.-4.9.), потребовалось несколько недель. Программирование подобной системы традиционным способом заняло бы не меньше полугода.

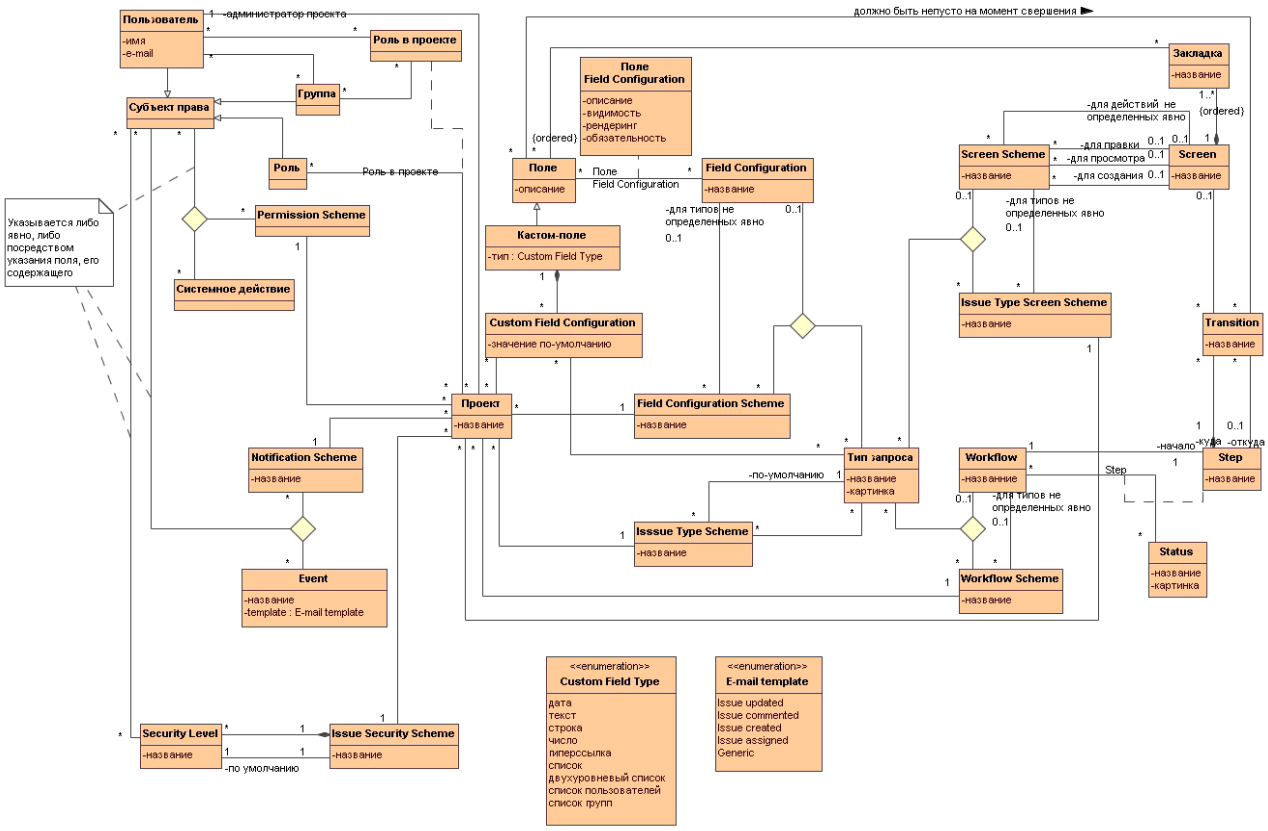


Рис. 4.10. Схема классов для настройки JIRA.

Ключевыми понятиями в JIRA являются проекты и запросы. Проекты служат для группирования запросов, которые в них создаются. У каждого проекта есть администратор, назначающий исполнителей запросов. Запросы могут быть разного типа и иметь подзадачи, а так же могут быть связанными с другими запросами.

Проект имеет сайт, схему рассылки нотификаций (уведомляющих сообщений), гибкую схему контроля доступа.

Каждому типу запроса можно сопоставить свою схему движения, свой вид экрана ("собрать" требуемый экран из имеющихся компонентов). JIRA также формирует отчеты по каждому проекту.

Применительно к процессу управления разработкой СППР, запросы имеют следующие типы: Новый продукт, Ошибка, Отчет. При описании каждого типа запроса имеется возможность управления набором полей. На рисунке 4.11. приведен пример запроса типа Ошибка.

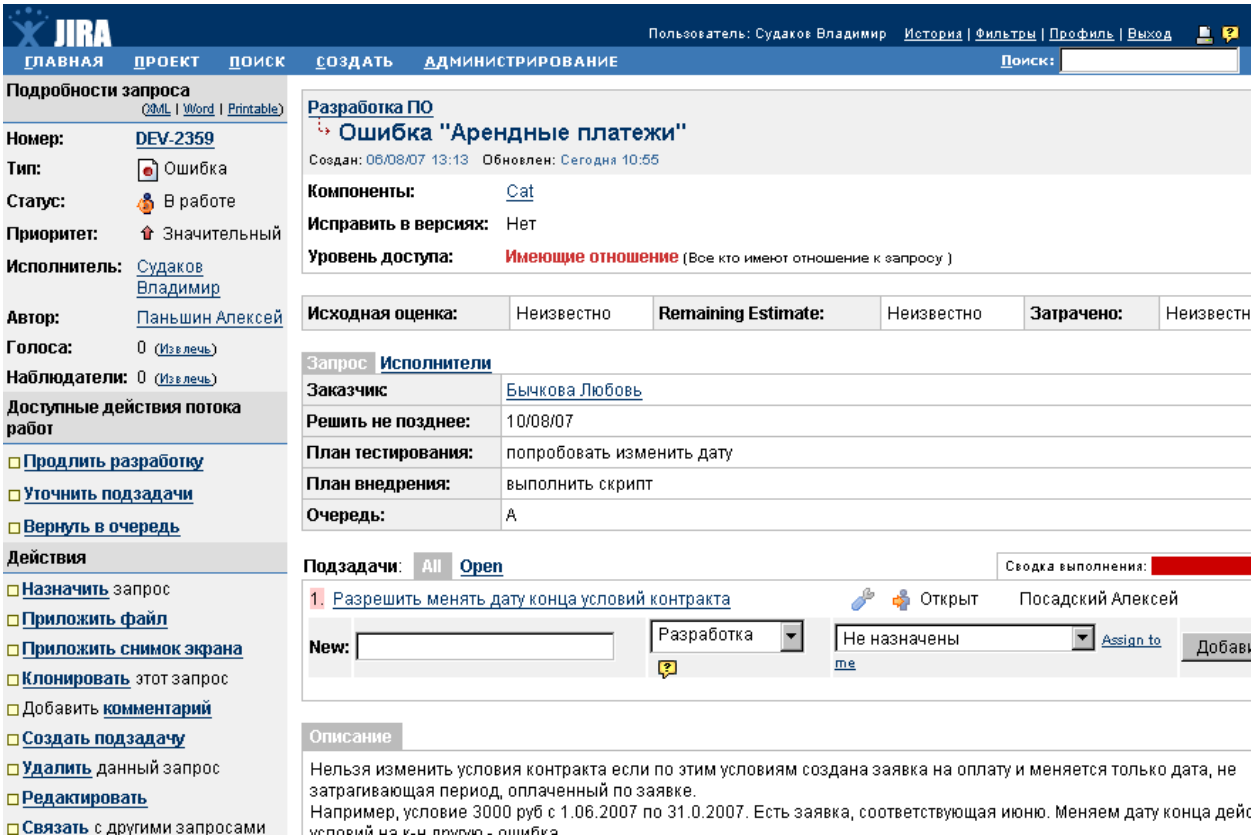


Рис. 4.11. Запрос в системе JIRA.

JIRA позволяет находить задания по всем критериям и по пользовательским полям, создавать фильтры, которые можно сделать общедоступными, а также организовать автоматическую рассылку результатов работы фильтров членам рабочей группы.

Запросы JIRA в каждый момент времени имеют определенный статус. Возможные действия с запросами, имеющими тот или иной статус, определяются встроенной системой управления движением запросов. Схема движения запроса может быть своя для каждого отдела, проекта, типа. Схема движения редактируется встроенным редактором. Редактируя движение запроса, создавая новые статусы и определяя возможные действия, можно организовать работу произвольного уровня сложности. Движение запроса можно сделать зависимым от условий, применять логику И/ИЛИ, выполнять определенны действия на каждом этапе (см. рисунок 4.12.).

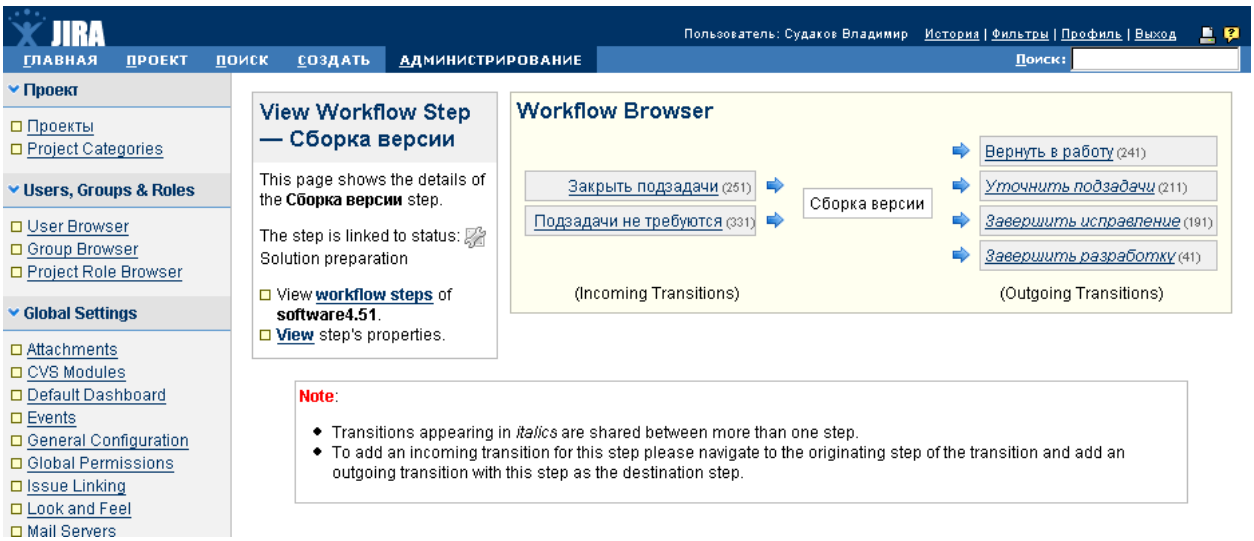


Рис. 4.12. Редактирование схемы движения запроса.

Для аналитических целей JIRA создает карту проекта, позволяет просматривать загрузку каждого пользователя и делает многое другое для эффективного управления проектами. Также имеется целый ряд необходимых стандартных отчетов:

- нерешенные высокоприоритетные запросы;
- количество запросов созданных одним пользователем;
- среднее время решения запроса;
- отношение реального и заданного времени решения запросов;
- количество запросов созданных в день, неделю, месяц, год ;
- популярные запросы (набравшие большинство голосов);
- запросы, назначенные для указанной версии или этапа;
- запросы, имеющие определенные статус;
- запросы, имеющие определенный приоритет;
- отчет по использованию времени;
- отчет о затратах времени на выпуск определенной версии;
- отчет о загрузке разработчиков.

Кроме стандартных отчетов, JIRA позволяет создавать пользовательские отчеты.

JIRA позволяет управлять видом специальной стартовой страницы, называемой приборной панелью (см. рисунок 4.13.).

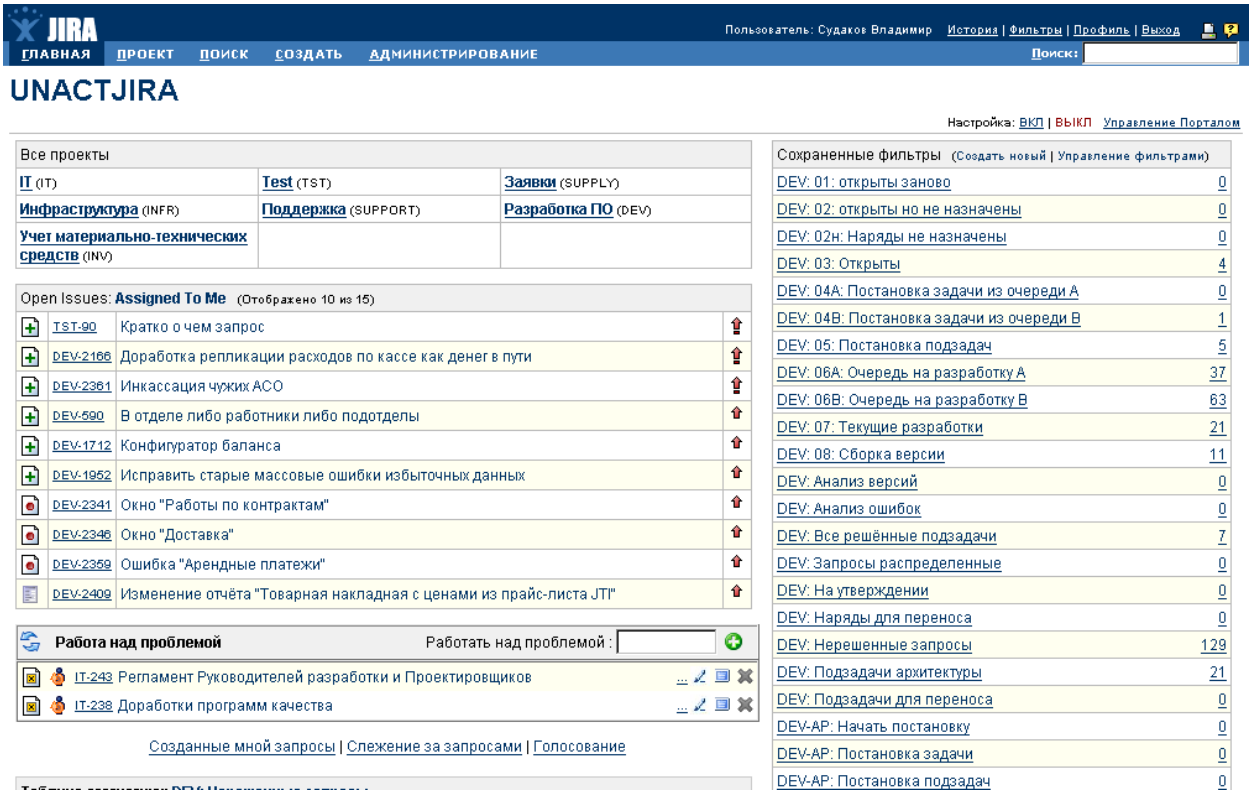


Рис. 4.13. Стартовая страница системы JIRA.

На рисунке 4.13. отображен ход выполнения проектов и содержатся ссылки для быстрого доступа ко всем часто используемым функциям, отчетам и запросам:

- список запросов, назначенных текущему пользователю;
- список сохраненных фильтров;
- статистика проекта;
- статистика фильтра;
- список проектов.

Работа с JIRA заключается в следующем:

- руководитель проекта вводит задание. Задание имеет тип. Предположим, что руководитель ввел задание на выполнение определенной работы. JIRA рассылает уведомления о созданном запросе всем членам группы, работающей над запросом. Запрос приобретает статус "открытый" и "не назначенный";
- руководитель назначает исполнителя или кто-то из рабочей группы, имеющий соответствующие права, сам берет себе этот запрос. JIRA рассылает уведомления - исполнителю о назначенном ему запросе, членам рабочей группы о происшедшем событии. Запрос приобретает статус "назначенный";
- назначенный исполнитель приступает к выполнению и запрос приобретает статус "в работе";

- исполнитель работает над заданием и вводит результаты работы - комментирует запрос, вводит проработанное время, присоединяет файлы, просматривает файлы, введенные другими, просматривает файлы, относящиеся к задаче из хранилища CVS;
- все важные действия исполнителя протоколируются и о них получают сообщения члены рабочей группы, руководители проекта и другие заинтересованные пользователи, имеющие на это право;
- исполнитель, закончив работу, присваивает запросу статус "решено" (но еще не "закрыто"). Система рассылает соответствующе сообщения. Наступает черед группы обеспечения качества проверить работу и, если все хорошо, то запросу присваивается статус "закрыто";
- руководитель проекта видит графики выполнения проектов в целом, следит за уровнем загрузки членов команды, видит просроченные запросы.

Описанные выше свойства системы JIRA позволили создать автоматизированную систему управления процессом разработки приложений поддержки решений для группы компаний ЮНЭКТ [142-144].

4.8. Критерий эффективности разработки программного обеспечения СППР

Эффективность управления разработкой определяется степенью удовлетворенности пользователей и заказчиков СППР, что достигается как за счет хорошей организации работы, так и за счет качественной работы исполнителей. Таким образом, объектами управления выступают отдельные исполнители (проектировщики, программисты, группа тестирования и др.) и их непосредственные руководители. Наиболее обобщенно объекты управления можно оценивать по следующим составляющим:

1. Ответственность.
2. Качество результатов.
3. Загруженность.
4. Стремление к развитию.

Ответственность – это способность объектов управления брать на себя обязательства и выполнять их в согласованные сроки. При управлении необходимо прогнозировать, учитывать и минимизировать возможные отклонения и иметь механизм согласования сроков.

Качество – это совокупность свойств объекта, обусловливающих его способность удовлетворять заданным потребностям в соответствии с его назначением. Качество выступает обычно как ограничение. Т.е. при любой разработке продукты должны иметь качество не ниже заданного уровня.

Загруженность – характеристика аналогичная пропускной способности исполнителя или коллектива исполнителей. Она определяется средней очередью невыполненных заданий и простоем исполнителей.

Стремление к развитию - способность объектов управления повышать свою квалификацию, вносить предложения по улучшению эффективности разработки СППР.

Для оперативного управления разработкой надо видеть текущую ситуацию и иметь возможность задать граничные условия для объектов управления. Для наблюдения за текущей ситуацией достаточно возможностей системы JIRA или аналогичной. Граничные условия накладываются на состояния запросов. Например, описание проекта решения проблемы пользователя на понятном ему языке архитекторы СППР должны подготовить в течение четырех дней для обычных проблем и в течение трех дней для проблем с повышенным приоритетом. Все эти условия описываются в регламентах работы. Регламент – это аналог должностной инструкции, в нем определены действия, порядок их исполнения и сроки. Регламенты участвуют в системе мотивации сотрудников. Мотивация, как правило, материальная (штрафы и премии).

Пример регламента:

Сроки совершения действий архитекторами.
Вводится в действие с 10 августа 2007 года.
Для чрезвычайных и угрожающих приоритетов:
Срок нахождения запроса из очереди A в состоянии «Открыт» и «Постановка задачи»:
не больше 1-го рабочего дня.
Срок нахождения запроса в состоянии «Открыт» и «Постановка задачи» для очереди B:
не больше 3-х рабочих дней.
Для остальных приоритетов:
Срок нахождения запроса в состоянии «Открыт» и «Постановка задачи» для очереди A:
не больше 2-х рабочих дней.
Срок нахождения запроса в состоянии «Открыт» и «Постановка задачи» для очереди B:
не больше 4-х рабочих дней.
Срок нахождения запроса в состоянии «Открыт» и «Постановка задачи» для очереди C:
не установлен.
При последующих входах в указанные состояния на время меньше или равное 4-м часам с момента последнего выхода из указанных состояний срок не продлевается.
При последующих входах в указанные состояния из состояния "Анализ проблемы" на время более 4-х часов срок исчисляется снова с момента последнего входа:
для очереди A не больше 1-го рабочего дня;

для очереди В не больше 3-х рабочих дней.

При первом входе в указанные состояния из состояния "На утверждении" на время более 4-х часов срок исчисляется снова с момента последнего входа следующим образом: для задач очереди А - добавляется один рабочий день, для остальных - два рабочих дня. При последующих входах в указанные состояния из состояния "На утверждении" на время более 4-х часов срок исчисляется снова с момента последнего входа следующим образом: добавляется один рабочий день.

При последующих входах в указанные состояния из состояния "Постановка подзадач" назначается срок равный дате последнего входа плюс один рабочий день.

Нарушение сроков влечёт штраф в размере XXX рублей, с каждого Архитектора, назначенного в просроченном запросе.

Каждый последующий рабочий день просрочки влечёт штраф в размере XXX рублей, с каждого Архитектора, назначенного в просроченном запросе.

Для контроля исполнения регламентов и фиксации их нарушений необходимо разработать систему, которая бы:

1. Предупреждала о предстоящих нарушениях регламента.
2. Фиксировала нарушения регламента.
3. Позволяла бы автоматизировать расчет премии по результатам выполнения заданий.

Для фиксации нарушений регламента разработана программа, которая с заданным временным интервалом проверяет, не появились ли нарушения, и фиксирует их в БД системы JIRA. На рисунке 4.14. приведен фрагмент схемы БД для описания правил регламента и фиксации его нарушений. На ней показано, как определяется, в каком проекте (project), для какого состояния (issuestatus), для каких типов запросов (issuetype) действуют заданные регламенты (reg_rule). В таблице reg_violation фиксируются нарушения регламентов. А в таблице reg_violation_user указаны исполнители, которые несут ответственность за зарегистрированные нарушения. После фиксации нарушения по протоколу SOAP программа вставляет соответствующий комментарий о нарушении и указывает срок, отведенный на исправление нарушения. Программа написана на C# и работает в виде Windows-сервиса. Обращение к БД JIRA для определения ситуаций нарушения регламента организовано с помощью ADO.NET.

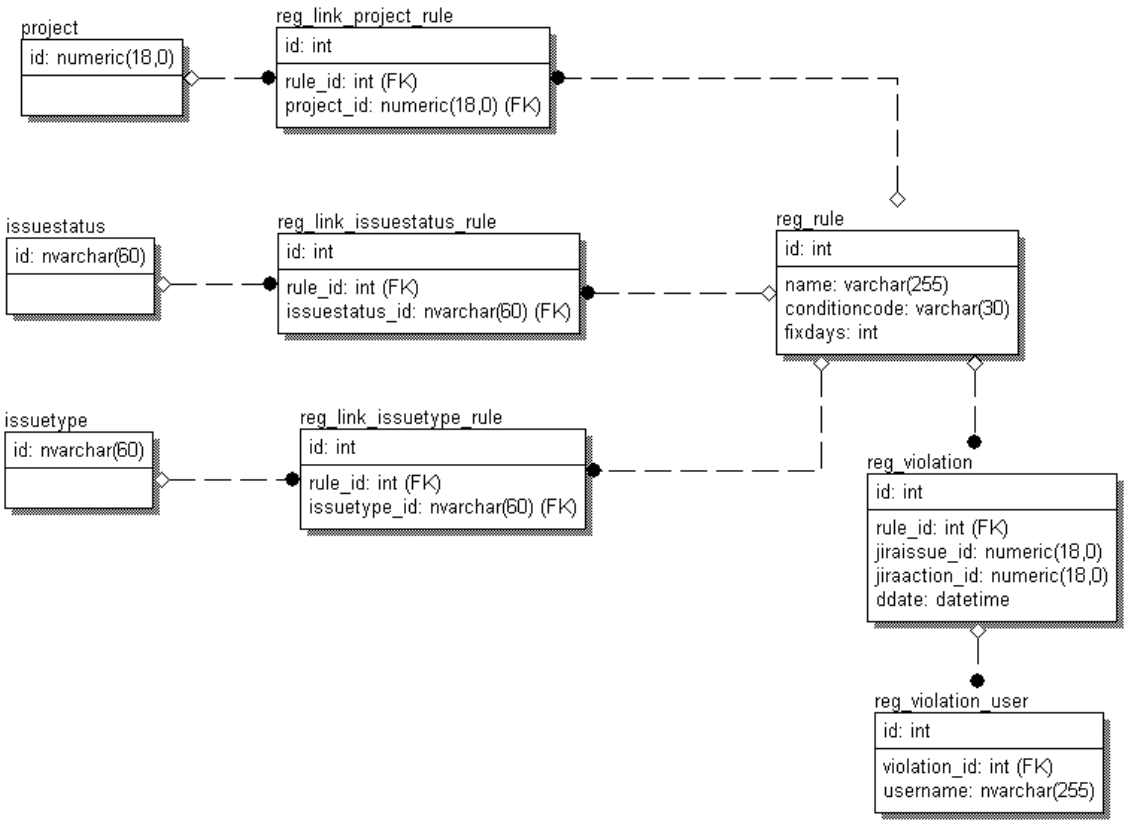
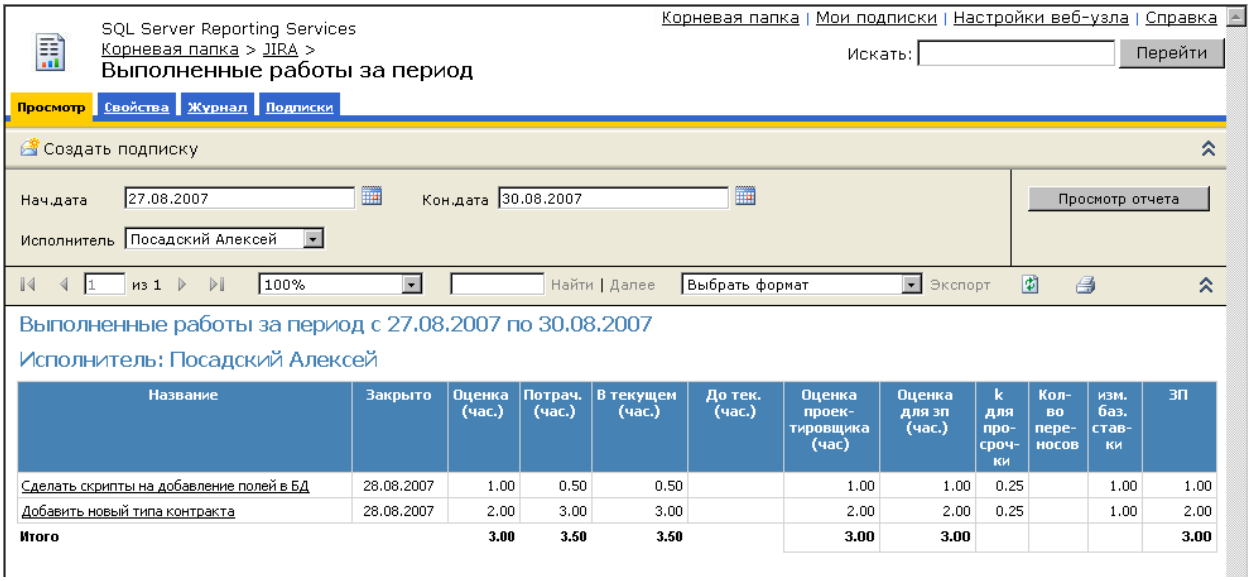


Рис. 4.14. Фрагмент схемы БД относящийся к регламентам.

Для предупреждения о сроках регламента и расчета премии по результатам выполнения регламентов удобно использовать отдельную отчетную систему. Она должна быть доступна всем заинтересованным лицам через Web интерфейс, позволять осуществлять рассылки отчетов. Оной из систем удовлетворяющей заданным требованиям является MS SQL Reporting Services. На рисунке 4.15. приведен отчет, который показывает расчет зарплаты для разработчика.



Название	Закрыто	Оценка (час.)	Потрач. (час.)	В текущем (час.)	До тек. (час.)	Оценка проек-тировщика (час)	Оценка для зп (час.)	k для про-сроч-ки	Кол-во пере-носов	изм. баз. став-ки	ЗП
Сделать скрипты на добавление полей в БД	28.08.2007	1.00	0.50	0.50		1.00	1.00	0.25		1.00	1.00
Добавить новый типа контракта	28.08.2007	2.00	3.00	3.00		2.00	2.00	0.25		1.00	2.00
Итого		**3.00**	**3.50**	**3.50**		**3.00**	**3.00**				**3.00**

Рис. 4.15. Расчет зарплаты разработчика ПО.

Однако система регламентов, способствуя показателям «Ответственность» и «Загруженность», не способствует развитию системы по показателям «Качество» и «Развитие». Наоборот, в их части она является демотивирующим фактором. При отсутствии механизмов постоянного контроля они могут снижаться. Для принятия решений в пространстве показанных показателей удобно воспользоваться системой поддержки многокритериальных решений СППР DSS/UTES [28-31]. Она ориентирована на конкретного пользователя, инвариантна к предметной области и, в отличие от большинства систем поддержки решений, работает с лексическими показателями.

Исходя из вышеизложенного подхода, нами в ГК ЮНЭКТ создана система отчётов, применение которой оказалось достаточно эффективной при оперативном управлении разработками приложений поддержки решений [143].

4.9. Разработка OLAP системы для мониторинга процесса разработки СППР

Помимо оперативных решений необходимо принимать и стратегические решения. Для этого необходимо анализировать большой объем информации за большой период времени. Например, какова тенденция длинны очереди на разработку, достаточно ли ресурсов, как долго обрабатываются запросы в разрезе проектов и исполнителей, определять «узкие» места в процессе разработки СППР. Эти задачи можно решать и средствами, которые есть в JIRA и в Reporting Services, но это не рационально по ряду причин:

1. Высокая трудоемкость создания аналитических отчетов и программирования всех необходимых фильтров и правил агрегирования.
2. Большие вычислительные нагрузки на систему оперативного управления при каждом запуске требуемого аналитического отчета.

Существуют OLAP технологии построения многомерных систем, которые позволяют решить задачи анализа и стратегического управления с меньшими трудозатратами. Одной из наиболее распространенных систем для этих целей является MS Analysis Services.

Для определенности поясним базовые понятия OLAP, как они понимаются на сайте Microsoft Technet (http://technet.microsoft.com/ru-ru/library/ms175680.aspx):

> *Измерение — это группа атрибутов. Атрибуты собираются из столбцов в одной или более таблицах-источниках. Например, измерение «Покупатель» может включать атрибуты «Имя покупателя», «Пол покупателя» и «Город покупателя» и может разрешить меры в кубе для анализа по «Имени покупателя», «Полу покупателя» и «Городу покупателя». Атрибуты внутри каждого измерения могут быть организованы в иерархии для предоставления путей анализа.*

Мера — это значение по транзакции или измерение, полученное по таблице фактов, которое пользователь может захотеть статистически обработать. Меры получают данные из столбцов в одной или более исходных таблиц и группируются в группы мер.

Куб — это набор связанных мер и измерений, которые используются для анализа данных.

Применительно к задаче мониторинга процессов разработки СППР в качестве примера рассмотрим две меры:

1. Потраченное на разработку время («Потрачено»).
2. Количество переходов запроса между состояниями («Кол-во переходов»).

Примеры измерений:

1. Проект – проект из системы JIRA.
2. Приоритет – приоритет запроса.
3. План – по плану или вне плана решался запрос.
4. Статус старый – статус, в котором запрос находился.
5. Статус новый – статус, в который запрос перешёл.
6. Исполнитель – ответственный за запрос.

Данные загружаются туда из различных источников, в частности, из системы JIRA. Фрагмент схемы загружаемых данных показан на рисунке 4.16.

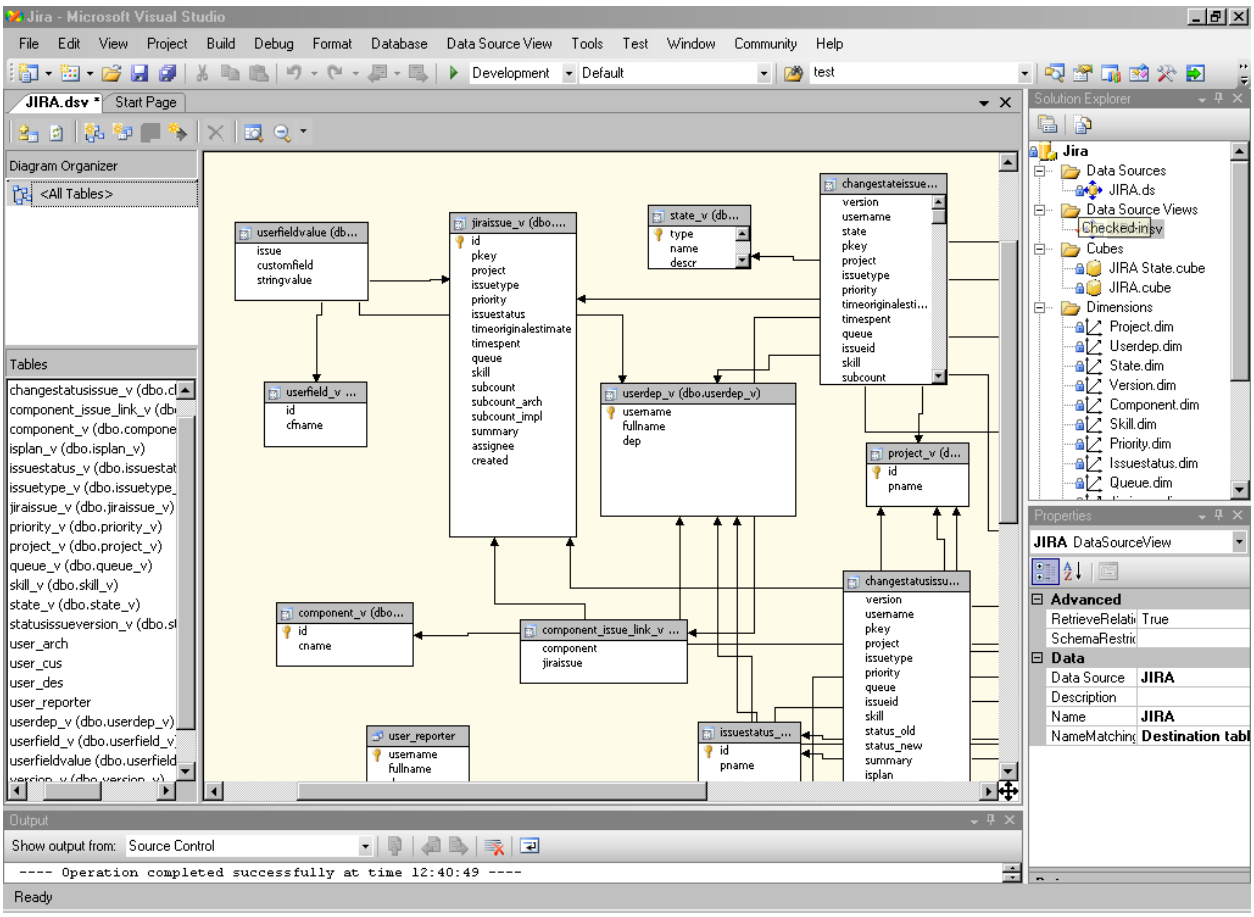


Рис. 4.16. Работа с источниками данных для куба.

После создания единого представления данных необходимо на их основе разработать измерения и один или несколько кубов содержащих требуемые меры. Измерения могут содержать несколько уровней иерархии. Например, у исполнителей запросов на разработку ПО 2 уровня: верхний уровень - отдел (Dep), нижний уровень - человек (Userdep). Проектирование измерения «Исполнители» показано на рисунке 4.17. Проектирование куба по данным из системы JIRA показано на рисунке 4.18.

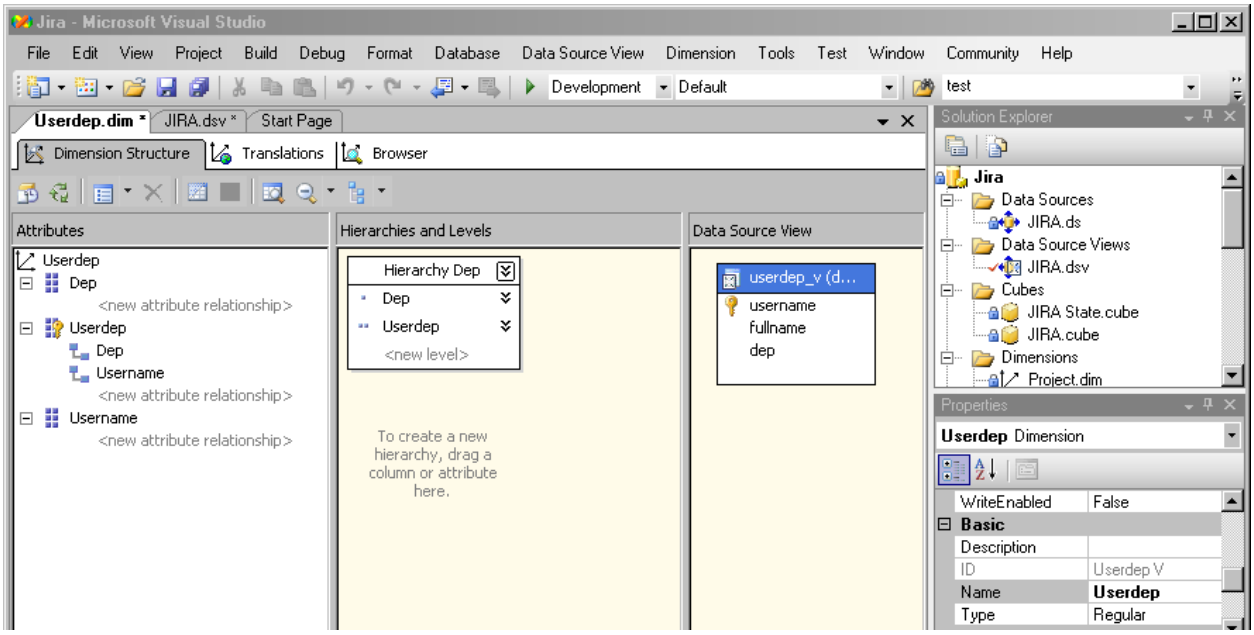


Рис. 4.17. Проектирование измерения «Исполнитель».

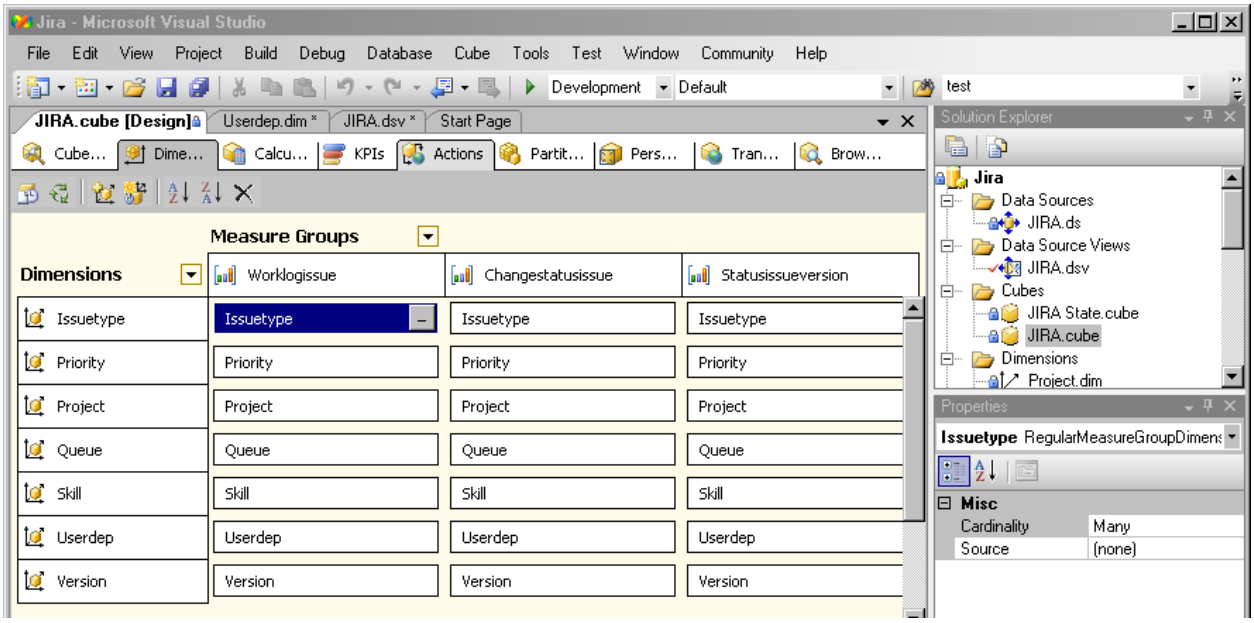


Рис. 4.18. Проектирование куба.

MS Analysis Services позволяет задавать перевод всех названий объектов на произвольное количество языков. Поэтому при просмотре созданного куба, можно использовать русский язык, а при проектировании английский.

Удобной и широко распространенной программой для просмотра кубов является ProClarity. Она позволяет работать не только с MS Analysis Services, но и с другими OLAP системами. С ней можно работать и через Web и через «толстого» клиента. ProClarity позволяет пользователям самостоятельно строить по кубам отчеты и запоминать их на специальном отчетном сервере. На рисунке 4.19. показано окно работы пользователя с кубами.

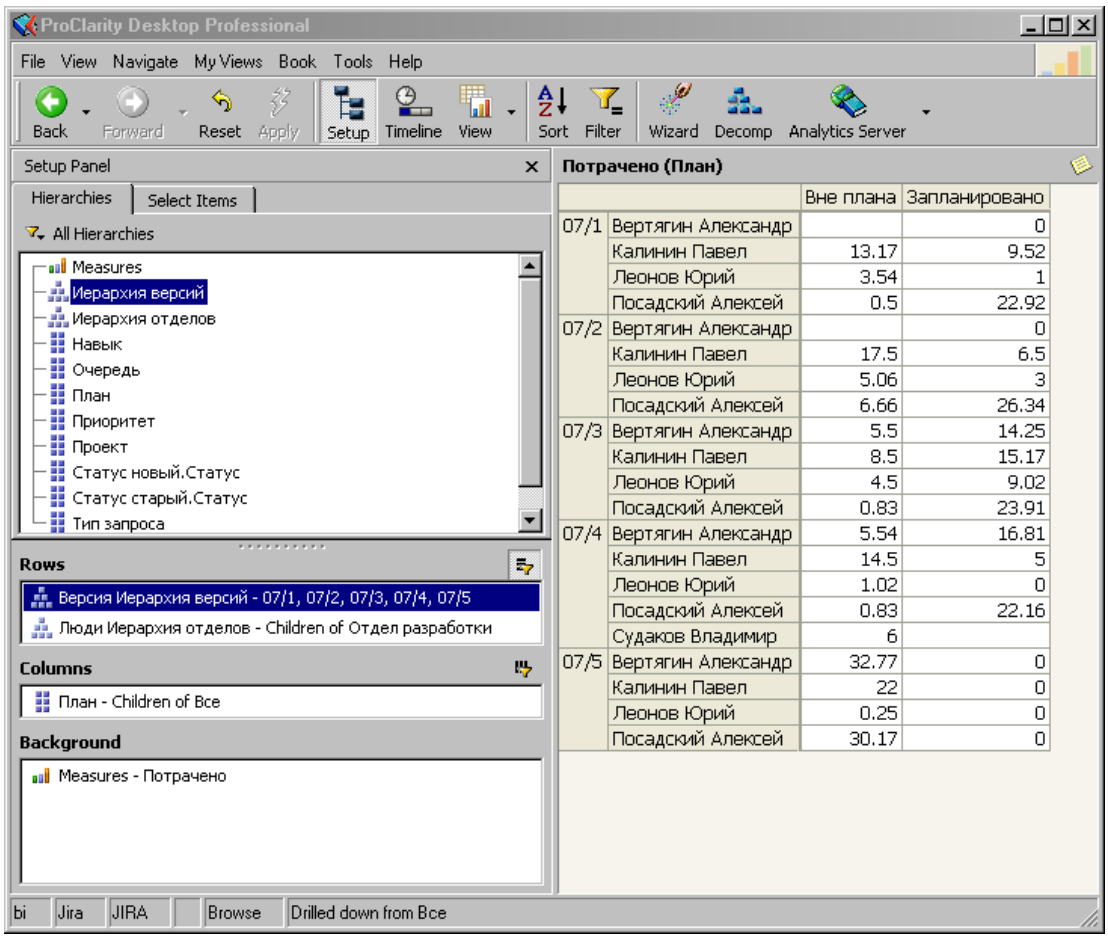
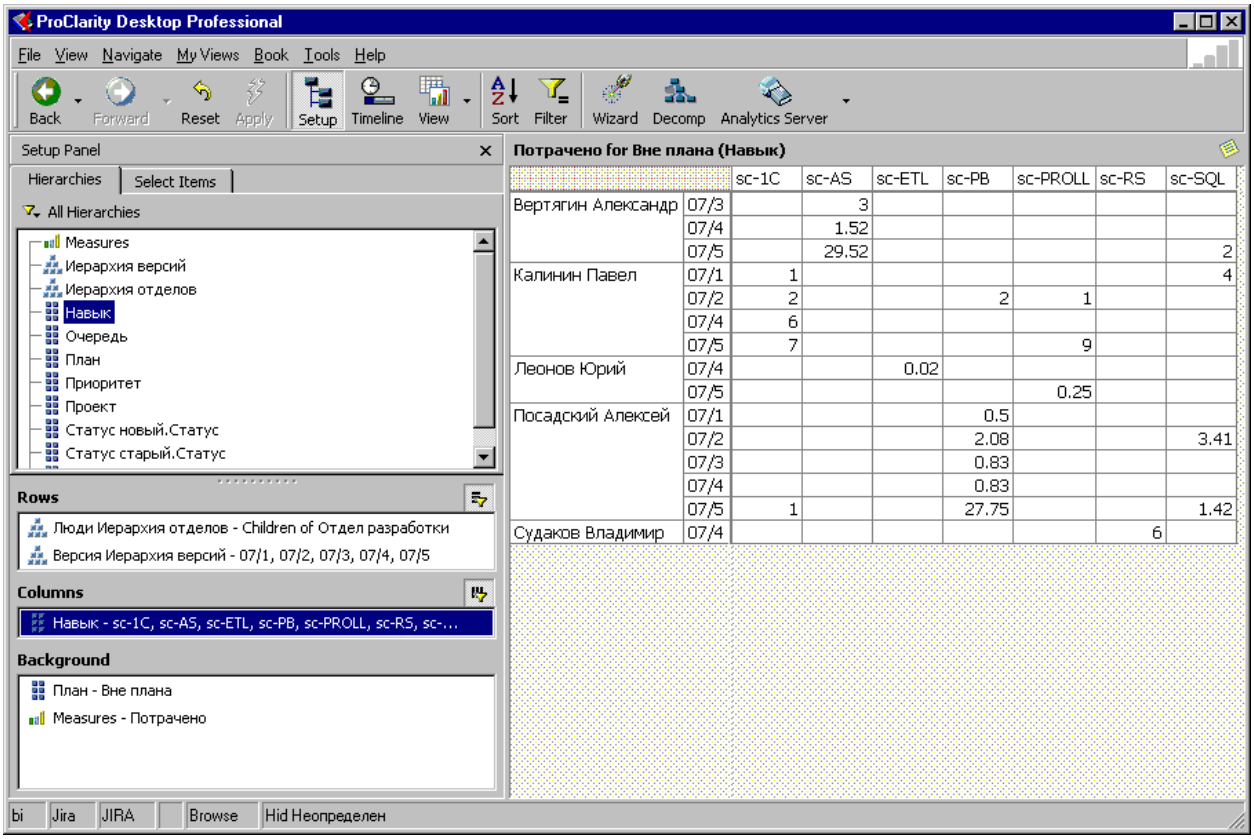


Рис. 4.19. Просмотр данных в кубе.

Существенным преимуществом многомерной базы данных перед традиционными является возможность раскрытия любой информации по произвольному числу измерений. Например, глядя на вышеприведенный рисунок, руководитель может проанализировать, как внеплановые задачи распределяются по разработчикам и навыками (см. рисунок 4.20.). В результате можно понять, каких специалистов сейчас недостаточно.



		sc-1C	sc-AS	sc-ETL	sc-PB	sc-PROLL	sc-RS	sc-SQL
Вертягин Александр	07/3		3					
	07/4		1.52					
	07/5		29.52					2
Калинин Павел	07/1	1						4
	07/2	2			2	1		
	07/4	6						
	07/5	7				9		
Леонов Юрий	07/4			0.02				
	07/5					0.25		
Посадский Алексей	07/1				0.5			
	07/2				2.08			3.41
	07/3				0.83			
	07/4				0.83			
	07/5	1			27.75			1.42
Судаков Владимир	07/4						6	

Рис. 4.20. Просмотр внеплановых разработок в кубе JIRA за месяц в разрезе исполнителей и навыков.

Применение описанных выше OLAP технологий показало свою эффективность при управлении разработкой СППР.

4.10. Моделирование процесса разработки СППР

4.10.1. Цели моделирования

При создании, эксплуатации, сопровождении, модернизации ПО СППР необходимо обеспечить руководителя вероятностно-временными оценками производительности сотрудников занятых при структурных и параметрических изменениях процесса разработки ПО СППР.

Под вероятностно-временными оценками производительности подразумевается:

1. Оценки времени обслуживания заявок разработку и модернизацию СППР:
   1.1. Гистограмма времени обслуживания заявки в системе.
   1.2. Оценка МО времени обслуживания заявок в системе в зависимости от приоритета.
   1.3. Оценка МО времени обслуживания заявок в системе в зависимости от типа.
2. Вероятность обслуживать заявку более критического времени.
3. Вероятность занятости исполнителя на стадиях обслуживания.

4. Оценка МО число заявок в системе на конец недели.

Под структурными и параметрическими изменениями процесса разработки понимается изменение количества исполнителей (добавление/удаление сотрудников) и перераспределения времени работы исполнителей на стадиях обслуживания. Для достижения этих целей был разработан программный продукт «Имитационная модель процесса разработки (ИМПР)». Этот программный продукт был успешно внедрен в группе компаний ЮНЭКТ как подсистема АСУ разработки программного обеспечения (АСУ РПО).

4.10.2. Конструирование имитационной модели

В подсистеме ИМПР моделированию подлежит только отдел разработки ПО, поскольку именно его деятельность необходимо оценить. В этом случае отдел ТП выступает в роли окружающей среды по отношению к отделу разработки ПО. Таким образом, в подсистеме ИМПР заявка может обслуживаться на двух стадиях обслуживания: «Архитектура» и «Разработка».

Переход заявки из стадии «Архитектура» в «Аналитика» (перемещение по переходу «Уточнить описание») в подсистеме ИМПР рассматривается как покидание заявкой отдела разработки ПО в том смысле, что при ее последующем возвращении из «Аналитики» в «Архитектуру» будет считаться, что в отдел разработки ПО вошла новая заявка. Данное упрощение позволило отказаться от учета в подсистеме ИМПР стадии «Аналитика», поскольку отпала необходимость учитывать пребывание заявки на этой стадии. Основаниями для данного упрощения явилось построение ИМПР, ориентированной исключительно на отдел разработки ПО и отсутствие статистики о количестве заявок, вошедших на стадию «Архитектура» из «Аналитика», предварительно побывав на стадии «Архитектура».

Переход «Уточнить подзадачи» в подсистеме ИМПР не учитывается. Данное упрощение позволило уменьшить возможное количество переходов в ИМ. Заявка, попавшая на стадию «Архитектура» из стадии «Разработка», находится в этой стадии недолго, т.к. обычно требуется уточнить некоторые технические детали у сотрудников отдела ТП. Так же отсутствует статистика о количестве таких переходов. Данные обстоятельства позволило отказаться от указанного перехода.

Переход заявки из стадии «Разработка» в «Тестирование» (перемещение по переходу «Завершить разработку») в ИМ рассматривается как окончание обслуживания заявки отделом разработки ПО. Последующее возвращение заявки в отдел разработки ПО (по переходу «Отправить на доработку» из стадии «Тестирование» или «Внедрение») в ИМПР не учитывается. Данное упрощение позволило отказаться от учета в ИМПР стадий

«Тестирование» и «Внедрение», поскольку отпала необходимость учитывать пребывание заявок в этих стадиях. Основаниями для данного упрощения явилось:

- построение ИМПР ориентированной исключительно на отдел разработки ПО;
- ошибки, при которых заявки возвращают в отдел разработки ПО, исправляют сразу же, как происходит переход. Если ошибки носят характер, который не позволяет сразу же их устранить, то создается новая заявка на разработку с целью устранения возникших ошибок;
- отсутствие статистики о количестве вернувшихся заявок в отдел разработки ПО.

В результате описанных упрощений, процесс выполнения заявок на разработку ПО для отдела разработки ПО приял вид, показанный на рисунке 4.21.

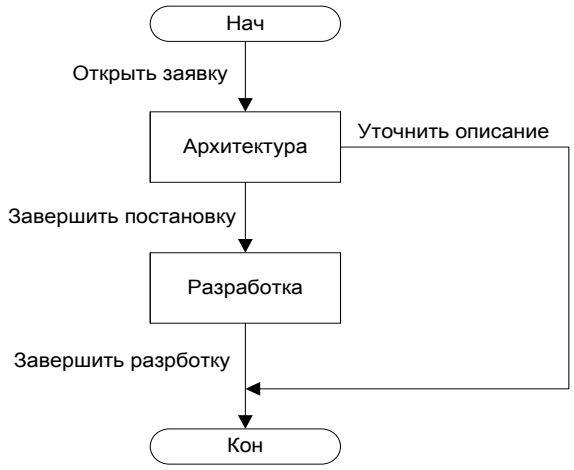


Рис. 4.21. Схема взаимосвязи между стадиями упрощенного процесса разработки ПО.

Данный процесс разработки ПО положен в основу подсистемы ИМПР.

При первом переходе из стадии «Аналитика» в «Архитектура» и при первом переходе из стадии «Архитектура» в «Разработка» для назначения заявки исполнителя применяется решающее правило, суть которого идентична в обоих случаях. При повторных переходах по указанным переходам заявка назначается на исполнение тому сотруднику, который был назначен при первом переходе.

Введем следующие обозначения:

i – идентификатор сотрудника отдела разработки ПО;

j – идентификатор типа заявки;

k – идентификатор приоритета;

n – номер стадии обслуживания;

S_{ijn} – опыт сотрудника с идентификатором i решать заявки с идентификатором j на стадии с номером n (определяется отношением числа решенных сотрудником с идентификатором i заявок с идентификатором j на стадии с номером n к общему числу решенных заявок с идентификатором j всеми сотрудниками на стадии с номером n);

$L_{in}(t)$ – загрузка сотрудника с идентификатором i в момент времени t на стадии с номером n (определяется отношением длины очереди заявок сотрудника с идентификатором i на стадии с номером n в момент времени t к суммарной длине очередей всех сотрудников на стадии с номером n момент времени t);

KS_{kn} – вес, вносимый опытом сотрудника, для случая заявки с приоритетом k на стадии с номером n;

KL_{kn} – вес, вносимый загрузкой сотрудника, для случая заявки с приоритетом k на стадии с номером n;

Тогда идентификатор сотрудника, которому будет назначена заявка, определяется следующей формулой:

$$I = \min_i arg\left\{\max_i\left(S_{ijn}KS_{kn} + \left(1 - L_{in}(t)\right)KL_{kn}\right)\right\}$$

Внешний минимум используется для однозначного определения идентификатора сотрудника отдела разработки ПО, если вдруг окажется, что максимальное значение в фигурных скобках соответствует более чем одному сотруднику.

В таблице 4.2. приведена агрегация приоритетов, используемая в решающем правиле. В таблице 4.3. приведены коэффициенты *KS* и *KL*, используемые в подсистеме ИМПР.

Табл. 4.2.

Агрегация приоритетов для решающего правила.

Приоритет	**Агрегированный приоритет**
Чрезвычайный	Чрезвычайный
Угрожающий	Угрожающий
Значительный	Прочие
Малозначительный	
Незначительный	

Табл. 4.3.

Коэффициенты *KS* и *KL*.

Приоритет	**Коэффициенты архитектуры**		**Коэффициенты разработки**	
	KS	*KL*	*KS*	*KL*
Чрезвычайный	1	0	1	0
Угрожающий	0,7	0,3	0,7	0,3
Прочие	0,3	0,7	0,3	0,7

Перед проведением имитационных экспериментов подсистема сбора статистики собирает статистическую информацию о процессе разработки ПО.

Для проведения имитационных экспериментов требуется предварительно собранная статистическая информации по БД АСУ РПО. Для этого расширена существующая подсистема сбора статистики. Сбор статистики производится с 7 августа 2007 года с 15:45, поскольку в этот

момент процесс разработки ПО был изменен на тот, который используется по сегодняшний день.

Заявки в ИМПР могут покидать отдел разработки ПО после стадии «Архитектура», так и не попав на стадию «Разработка». Для определения данной частоты разработан запрос к БД АСУ РПО. Частота покидания отдела разработки ПО заявками после стадии «Архитектура» составляет 0,605.

Для определения частоты появления заявок определенного типа разработан запрос к БД АСУ РПО, который просматривает все заявки и определяет частоту появления заявок с определенным типом. В таблице 4.4. приведены результаты работы запроса. На рисунке 4.22. приведены те же результаты в графическом виде.

Табл. 4.4.

Типы заявок с указанием частоты появления заявок определенного типа.

Тип	Частота
1C	0,071
Cat	0,607
Payroll	0,031
ReportServer	0,075
Прочие	0,216

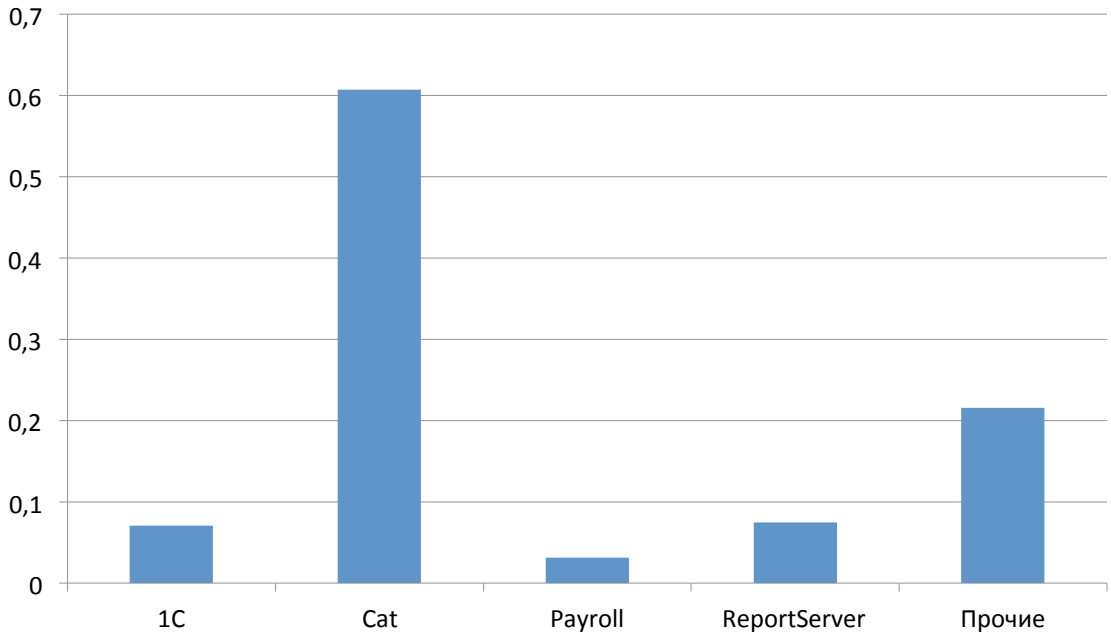


Рис. 4.22. Типы заявок с указанием частоты появления заявок определенного типа.

В таблице 4.5. определен перечень приоритетов заявок, актуальных для решающего правила. Этот же список актуален и для ИМПР. Для определения частоты появления заявок с определенным приоритетом разработан запрос к БД АСУ РПО, который просматривает все заявки и определяет частоту появления заявок с определенным приоритетом. В таблице 4.5.

приведены результаты работы запроса. На рисунке 4.23. приведены те же результаты в графическом виде.

Табл. 4.5.

Типы заявок с указанием частоты появления заявок определенного приоритета.

Тип	Частота
Чрезвычайный	0,025
Угрожающий	0,386
Прочие	0,589

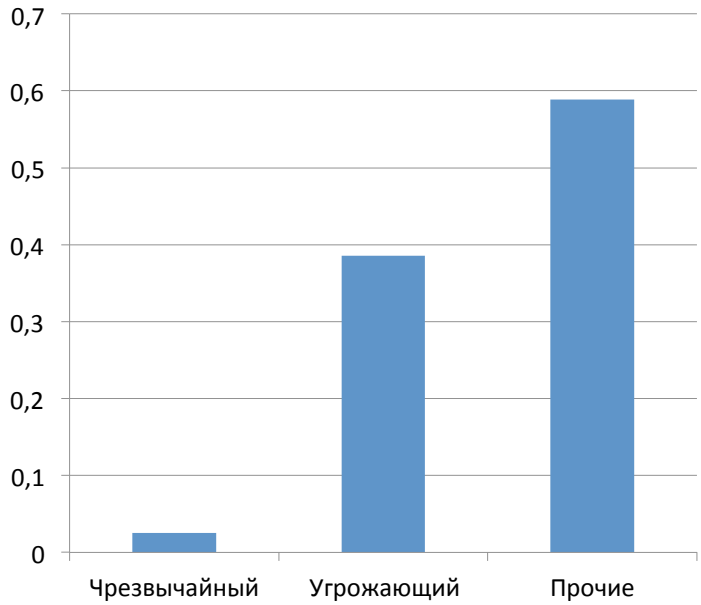


Рис. 4.23. Приоритеты заявок с указанием частоты появления заявок определенного приоритета.

Необходимо определить время, которое сотрудник работает на стадиях «Архитектура» и «Разработка». В АСУ РПО не предоставляется возможности непосредственно определить, сколько времени сотрудник проводит на той или иной стадии обслуживания. Для этой цели применяется следующий метод оценки, позволяющий определить, сколько времени сотрудник проводит на стадиях обслуживания:

1. Определить сколько заявок обслужил сотрудник на стадии «Архитектура».
2. Определить подзадачи каких заявок он обслуживал на стадии «Разработка», т.е. определить, какие запросы он обслуживал на стадии «Разработка».
3. Отношение числа обслуженных сотрудником заявок на стадии «Архитектура» к сумме обслуженных заявок на стадиях «Архитектура» и «Разработка» есть доля рабочего дня, которую сотрудник проводит на стадии «Архитектура».
4. Отношение числа обслуженных сотрудником заявок на стадии «Разработка» к сумме обслуженных заявок на стадиях «Архитектура» и «Разработка» есть доля рабочего дня, которую сотрудник проводит на стадии «Разработка».

5. Умножив полученные доли рабочего дня на продолжительность рабочего дня сотрудника, получаем время, которое сотрудник работает на стадиях «Архитектура» и «Разработка».

В таблице 4.6. приведены результаты работы запроса с указанием продолжительности рабочего дня сотрудника отдела разработки ПО. На рисунке 4.24. приведены те же результаты в графическом виде.

Табл. 4.6.

Оценка времени, которое сотрудник работает на стадиях «Архитектура» и «Разработка».

Сотрудник	**Продолжительность рабочего дня, час**	**Время работы на стадии «Архитектура», час.**	**Время работы на стадии «Разработка», час.**
Судаков В.А.	3	2,329	0,671
Калинин П.Е.	8	6,49	1,51
Посадский А.И.	8	4,759	3,241
Семухин М.В.	8	4,138	3,862
Куртов М.В.	8	1,552	6,448

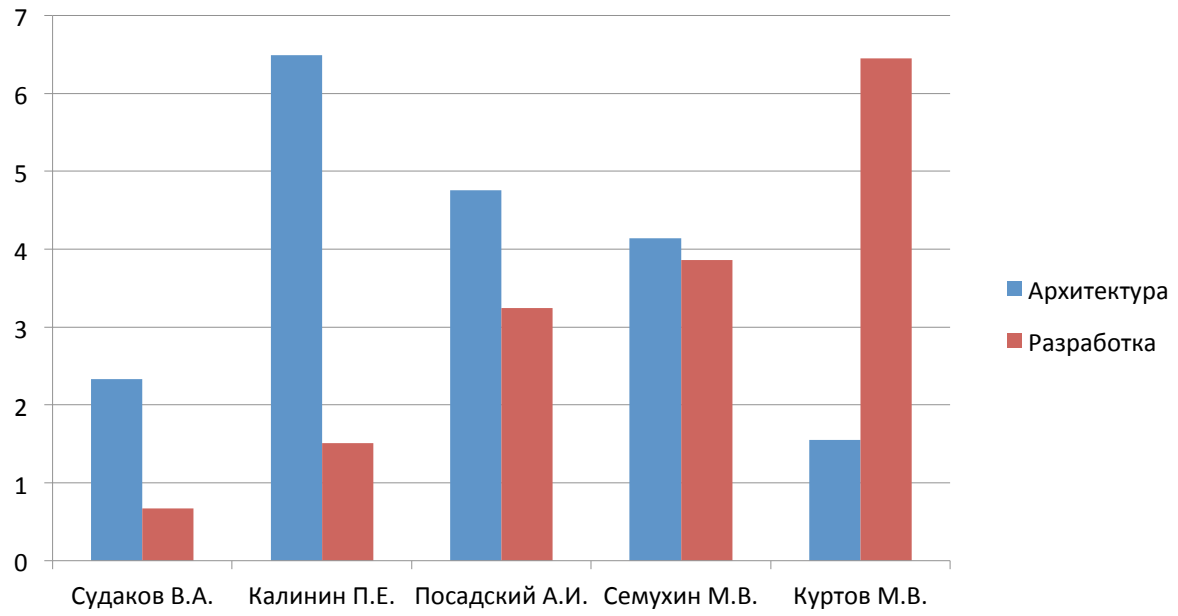


Рис. 4.24. Оценка времени, которое сотрудник работает на стадиях «Архитектура» и «Разработка».

В отделе разработки ПО сотрудники имеют разный опыт решения заявок различных типов. Что бы учесть этот факт в ИМПР по БД АСУ РПО собрана информация, сколько заявок (подзадач) и какого типа обслужил сотрудник на каждой стадии обслуживания. Используется следующая методика подсчета:

1. Подсчитать, сколько заявок определенного типа обслужил сотрудник на стадии «Архитектура».

2. Подсчитать, сколько подзадач определенного типа обслужил сотрудник на стадии «Разработка». Тип подзадачи соответствует типу заявки, которой принадлежит подзадача.
3. Отношение числа обслуженных сотрудником заявок с определенным типом к общему числу обслуженных заявок с тем же типом и есть опыт сотрудника решать заявки определенного типа на стадии «Архитектура».
4. Отношение числа обслуженных сотрудником подзадач с определенным типом к общему числу обслуженных подзадач с тем же типом и есть опыт сотрудника решать *заявки* определенного типа на стадии «Разработка».

В таблице 4.7. и на рисунке 4.25. приведены результаты работы запроса для стадии «Архитектура». В таблице 4.8. и на рисунке 4.26. приведены результаты работы запроса для стадии «Разработка».

Табл. 4.7.

Опыт сотрудников решать заявки с определенным типом на стадии «Архитектура».

Сотрудник	**Тип**				
	1C	**Cat**	**Payroll**	**ReportServer**	**Прочие**
Судаков В.А.	0,038	0,102		0,089	0,172
Калинин П.Е.	0,930	0,474	0,784	0,063	0,320
Посадский А.И.	0,032	0,248	0,176	0,165	0,189
Семухин М.В.		0,152	0,040	0,658	0,295
Куртов М.В.		0,024		0,025	0,024

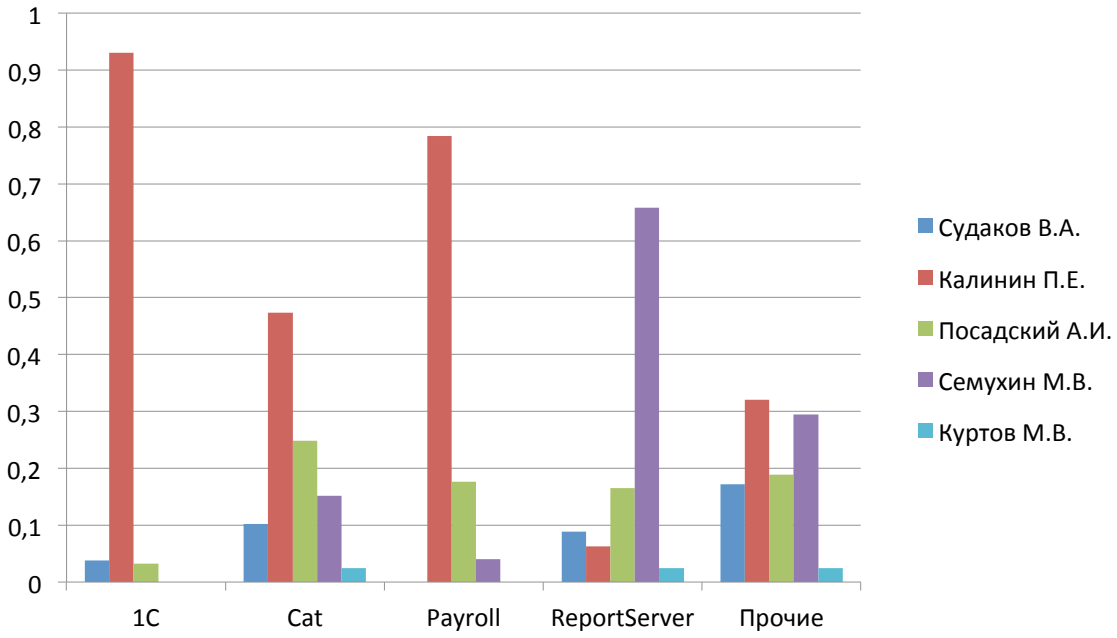


Рис. 4.25. Опыт сотрудников решать заявки определенного типа на стадии «Архитектура».

Табл. 4.8.

Опыт сотрудников решать заявки с определенным типом на стадии «Разработка».

Сотрудник	Тип				
	1С	Cat	Payroll	ReportServer	Прочие
Судаков В.А.	0,010	0,031		0,014	0,093
Калинин П.Е.	0,685	0,159	0,592		0,186
Посадский А.И.	0,269	0,351	0,347	0,088	0,240
Семухин М.В.	0,036	0,278	0,061	0,764	0,365
Куртов М.В.		0,181		0,134	0,116

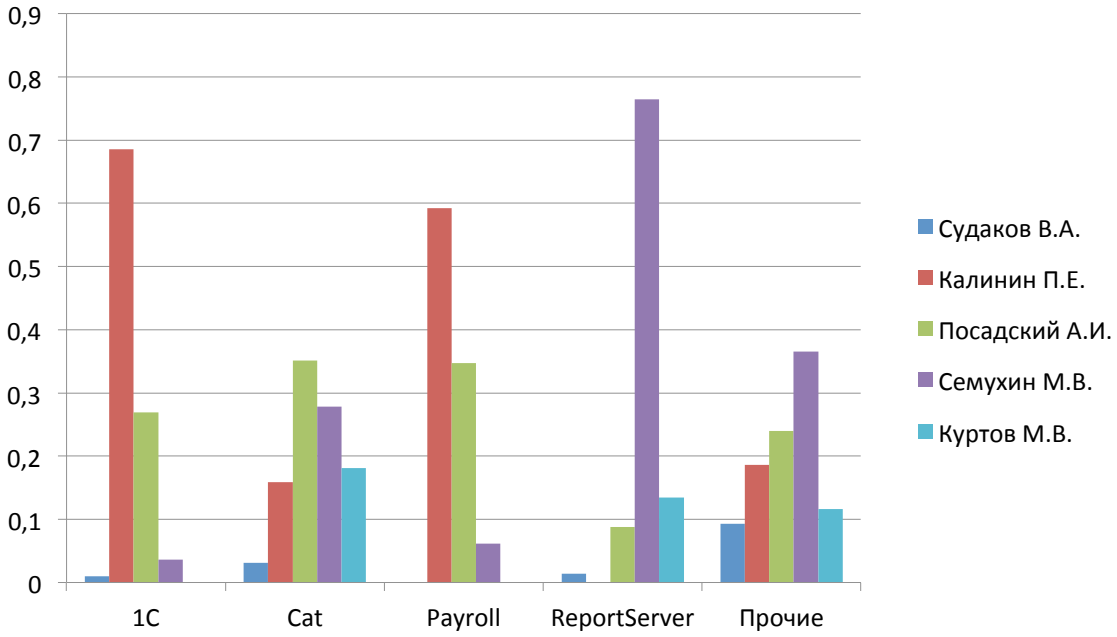


Рис. 4.26. Опыт сотрудников решать заявки определенного типа на стадии «Разработка».

Как было сказано в предыдущем разделе, сотрудники отдела разработки имеют разный опыт решать заявки различных типов. В связи с этим они так же тратят различное время на обслуживание заявок определенного типа.

Для определения интервалов времени между входами заявок в отдел разработки ПО разработан запрос к БД АСУ РПО. Данная статистика собиралась раздельно для будних и выходных дней, поскольку интенсивность входа заявок в эти дни различна. Закон распределения является многомерной функции переменного числа аргументов. Что бы в дальнейшем не моделировать случайные векторы традиционным методом (методом исключений), принято решение собирать статистику о реализациях реальных дней и хранить в БД моменты входа заявок в отдел разработки ПО каждого дня.

За основу построения структуры модели берется упрощенный процесс выполнения заявок на разработку ПО отделом разработки ПО. Описанный процесс хорошо определяется традиционным структурным представлением типовой математической моделью многофазной системы массового обслуживания (СМО) с выделенным источников заявок, ресурсных единиц

и очередей. Разработка модели происходит средствами системы имитационного моделирования Modelling, которая позволяет существенно упростить процесс разработки имитационных моделей СМО.

Связь элементов теории массового обслуживания с элементами реального процесса:

- заявки – заявки на разработку ПО, поступившие из отдела ТП (окружающая среда для СМО) в отдел разработки ПО (сама СМО);
- фаза «Архитектура» (стадия «Архитектура») – соответствует стадии «Архитектура» реального процесса;
- фаза «Разработка» (стадия «Разработка») – соответствует стадии «Разработка» реального процесса;
- ресурсные единицы на стадии «Архитектура» – сотрудники отдела разработки ПО (исполнители), работающие на стадии «Архитектура»;
- ресурсные единицы на стадии «Разработка» – сотрудники отдела разработки ПО (исполнители), работающие на стадии «Разработка».

На рисунке 4.27. приведена структурная схема ИМПР.

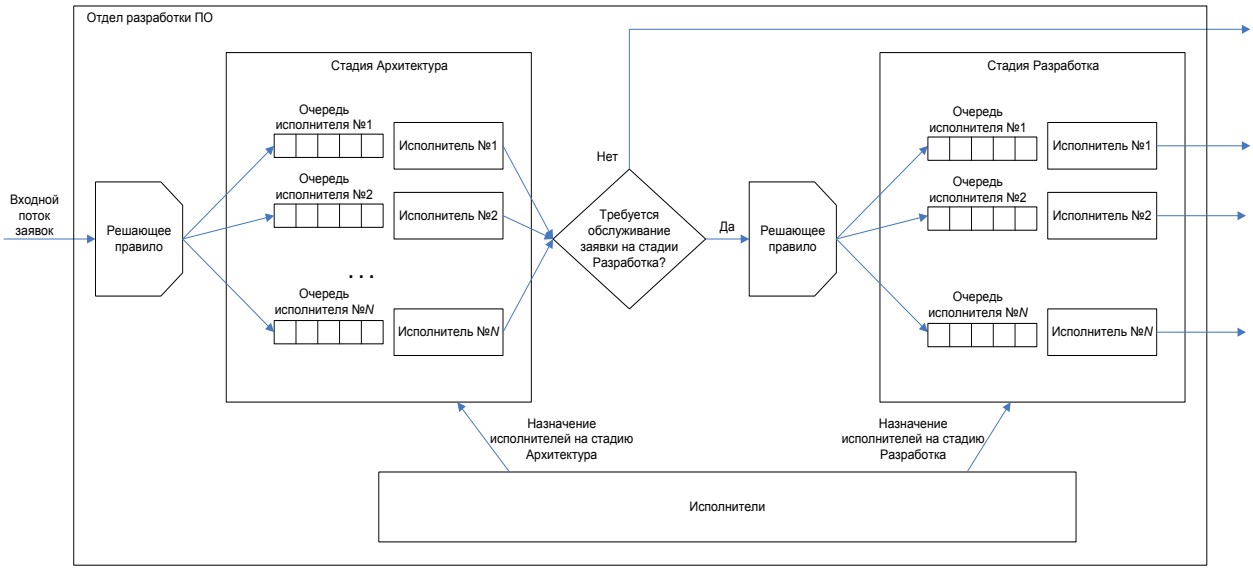


Рис. 4.27. Структурная схема ИМПР.

В СМО имеются исполнители, которые могут обслуживать заявки на обоих стадиях обслуживания. Перед началом моделирования известно, какое время каждый день исполнитель тратит на обслуживания заявок на стадии «Архитектура» и «Разработка». В начале каждого дня исполнитель работает на стадии «Архитектура», остальное время – на стадии «Разработка».

На вход СМО подается случайный входной поток со случайными и приоритетами. Войдя в систему, заявка попадает на вход решающего правила, которое в зависимости от опыта решать заявки на стадии «Архитектура» того типа, как у рассматриваемой заявки, и текущей их

загрузки, определяет какому исполнителю назначить заявку на обслуживание на стадии «Архитектура». Если исполнитель в данный момент работает на стадии и свободен, то заявка начинает обслуживание. В противном случае становится в очередь исполнителя. Время, которое заявка должна обслуживаться на стадии «Архитектура» тем исполнителем, который на это назначен, моделируется на основании собранной статистики. Обслуживание заявки может прерываться вследствие окончания работы исполнителя на стадии «Архитектура». В этом случае, после возобновления работы исполнителя на стадии «Архитектура» он продолжит обслуживать ту заявку, обслуживание которой прервалось. После того, как заявка полностью обслужится на стадии «Архитектура», случайным образом принимается решение о продолжении обслуживания заявки в СМО. Если принято отрицательное решение, то заявка покидает СМО.

В случае, если принято решение продолжить обслуживание, она попадает на вход решающего правила, которое в зависимости от опыта исполнителей решать заявки на стадии «Разработка» того типа, как у рассматриваемой заявки, и текущей их загрузки, определяет какому исполнителю назначить заявку на обслуживание на стадии «Разработка». Если исполнитель в данный момент работает на стадии и свободен, то заявка начинает обслуживание. В противном случае становится в очередь исполнителя. Время, которое заявка должна обслуживаться на стадии «Разработка» тем исполнителем, который на это назначен, моделируется на основании собранной статистики. Обслуживание заявки может прерываться вследствие окончания работы исполнителя на стадии «Разработка». В этом случае, после возобновления работы исполнителя на стадии «Разработка», он продолжит обслуживать ту заявку, обслуживание которой прервалось. После того, как заявка полностью обслужится на стадии «Разработка», она покидает СМО.

После того как исполнитель обслужит заявку он берет из очереди (если такая есть) самую приоритетную заявку. Если таких заявок несколько, то он берет ту, которая встала в очередь раньше. Если очередь пустая, то исполнитель простаивает на стадии обслуживания, пока ему не будет назначена новая заявка. Сказанное относится к обоим стадиям обслуживания.

У исследователя имеется возможность варьировать количество исполнителей в СМО, а так же перераспределять время, которое они должны работать на стадиях обслуживания. Делается это для того, что бы получить рациональное решение исходя из критериев моделирования.

4.10.3. Выбор класса имитационной модели

По отображению временной координаты процесса функционирования моделируемой системы проектируемая ИМ относится к классу динамических моделей. Динамические модели воспроизводят процесс функционирования моделируемой системы во времени [156].

По характеру представления процесса функционирования моделируемой системы во времени проектируемая ИМ относится к дискретным ИМ, поскольку отражает процесс функционирования моделируемой системы дискретно, по отдельно взаимосвязанным между собой моментами модельного времени, в которые происходят значимые с точки зрения целей исследования изменения состояния системы [139].

По учету стохастических факторов проектируемая ИМ относится к стохастическим, поскольку моделируемая система функционирует в стохастических условиях. Вероятностный характер имеют, например, такие процессы, как вход заявки на обслуживание на стадию «Архитектура», назначение приоритета и типы заявки.

Самым важным решением при выборе класса ИМ – было отнести ее к классу дискретных систем. На этот выбор повлияли:

- структурное представление модели как системы массового обслуживания;
- стремление сделать модель как можно проще в вычислительном смысле;
- цель исследования позволяет осуществить адекватную дискретизацию;
- имеющиеся в распоряжении разработчика инструментальные средства автоматизации имитационного программирования дискретных систем.

4.10.4. Критерии оценки результатов моделирования

В качестве критериев оценки результатов моделирования выбранные следующие критерии:

1. Время обслуживания заявок в системе:

1.1. timeInSystemAll – гистограмма времени обслуживания заявки в системе,

1.2. timeInSystemByPriorrity[j] – оценка МО времени обслуживания заявок в системе с *j*-м приоритетом,

1.3. timeInSystemByType[j] – оценка МО времени обслуживания заявок в системе с *j*-м типом.

2. ProbInSystemMoreCritical – вероятность обслуживать заявку более критического времени.

3. ProbLoadInArch, probLoadInDev – вероятность занятости исполнителя на стадиях обслуживания.

4. IssuesAtFriday – оценка МО числа заявок в системе на конец недели.

4.10.5. Модельные события

Принимая во внимание дискретность событий, происходящих в системе, будет разрабатываться ИМ в классе дискетных моделей с ориентацией на метод модельных событий.

Можно выделить следующий список модельных событий:

1. Вход заявки в систему.
2. Заявка полностью обслужилась на стадии «Архитектура».
3. Заявка полностью обслужилась на стадии «Разработка».
4. Начало работы исполнителей на стадии «Архитектура» (начало рабочего дня).
5. Начало работы исполнителя на стадии «Разработка» (завершение работы исполнителя на стадии «Архитектура»).
6. Завершение работы исполнителя на стадии «Разработка».
7. Завершение суток.

Взаимосвязь модельных событий показана на графе модельных событий (см. рисунок 4.28.).

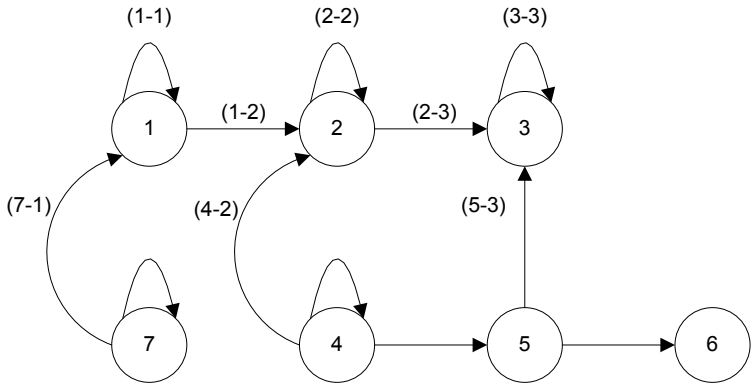


Рис. 4.28. Граф модельных событий.

Условия планирования событий:

(1-1) если в текущий день должны войти еще заявки;

(1-2) если выбранный для обслуживания исполнитель не занят запросом на стадии «Архитектура» и в данный момент работает на стадии «Архитектура»;

(2-2) если очередь к освободившемуся исполнителю на стадии «Архитектура» не пуста;

(2-3) если выбранный для обслуживания исполнитель не занят запросом на стадии «Разработка» и в данный момент работает на стадии «Разработка»;

(3-3) если очередь к освободившемуся исполнителю на стадии «Разработка» не пуста;

(4-2) если очередь исполнителя на стадии «Архитектура» не пуста;

(5-3) если очередь исполнителя на стадии «Разработка» не пуста;

(7-1) если в начавшийся день должны войти заявки.

Каждый переход в графе модельных событий планирует одно модельное событие, за исключением переходов из 4-го модельного события во 2-е и из 4-го в 5-е. Количество запланированных событий по переходу из 4-го модельного события во 2-е – равняется количеству исполнителей, для которых условие (4-2) принимает истинное значение. Количество запланированных событий по переходу из 4-го модельного события в 5-е равняется количеству исполнителей.

4.10.6. Атрибуты модельных событий

Модельное событие № 1 (Вход заявки в систему) имеет следующие атрибуты:

- время свершения события;
- код события;
- номер заявки;
- идентификатор приоритета заявки;
- идентификатор типа заявки.

Модельное событие № 2 (Заявка полностью обслужилась на стадии «Архитектура») имеет следующие атрибуты:

- время свершения события;
- код события;
- номер заявки;
- идентификатор исполнителя.

Модельное событие № 3 (Заявка полностью обслужилась на стадии «Разработка») имеет следующие атрибуты:

- время свершения события;
- код события;
- номер заявки;
- идентификатор исполнителя.

Модельное событие № 4 (Начало работы исполнителей на стадии «Архитектура») имеет следующие атрибуты:

- время свершения события;
- код события.

Модельное событие № 5 (Начало работы исполнителя на стадии «Разработка») имеет следующие атрибуты:

• время свершения события;

• код события;

• идентификатор исполнителя.

Модельное событие № 6 (Завершение работы исполнителя на стадии «Разработка») имеет следующие атрибуты:

• время свершения события;

• код события;

• идентификатор исполнителя.

Модельное событие № 7 (Завершение суток) имеет следующие атрибуты:

• время свершения события;

• код события.

4.10.7. Определение системы параметров

Параметры – это такие характеристики модели, которые задаются перед началом имитационных исследований очередного варианта системы и во время прогона или множества прогонов не изменяются.

Параметры, характеризующие моделируемую систему и внешнюю по отношению к ней среду, перечислены в следующем списке:

1. TP – время моделирования;
2. workDayStart – начало рабочего дня исполнителей;
3. durationDay – продолжительность суток;
4. criticalTime – критическое время;
5. leaveProbAfterArch – частота покидания СМО заявками после стадии «Архитектура» (см. п. **Ошибка! Источник ссылки не найден.**);
6. объект emp класса TEmp инкапсулирует перечень исполнителей с указанием времени, которое исполнитель работает на стадии «Архитектура» и «Разработка»;
7. объект issueType класса TPoligon инкапсулирует перечень типов заявок с частотой появления заявок определенного типа;
8. объект issuePriority класса TPoligonKoef инкапсулирует:

8.1. перечень приоритетов заявок с частотой появления заявок определенного приоритета;

8.2. коэффициенты приоритетов, используемые в решающем правиле для распределения заявок по исполнителям на каждой стадии обслуживания;

9. объект empSkill класса TEmpSkill инкапсулирует опыт исполнителей решать заявки с определенным типом на каждой стадии обслуживания;

10. объект empDelay класса TEmpDelay инкапсулирует законы распределения времени обслуживания заявок определенного типа для каждого исполнителя на каждой стадии обслуживания;
11. объект arrivalTimes класса TArrivalTimes инкапсулирует законы распределения интервалов времени между входами заявок в СМО в рабочие и выходные дни.

Параметры с 1-го по 5-й выполнены в виде модельных переменных действительного типа. Параметры с 6-го по 11-й выполнены в виде объектов соответствующих классов.

4.10.8. Переменные состояния

Разработка ИП велась с использованием системы имитационного моделирования Modelling [96, 97]. В связи с этим переменные состояния составлялись с ориентацией на Modelling. СИМ Modelling поддерживает переменные состояния двух видов:

1. Списковые структуры. Списковые структуры являются удобной формой задания переменных состояния ИМ со сложной специфической внутренней структурой. С их помощью удобно определять состояние различных типов очередей в ИМ СМО, задавать динамические изменяющиеся массивы моделируемых объектов и т.п. Списковые структуры в СИМ Modelling поддерживают различные типы организации (в том числе: очередь, стек и др.) и автоматизированный сбор статистики по их использованию.
2. Модельные переменные. Все модельные переменные в СИМ Modelling являются векторными (скалярные переменные определяются как вектор размером 1). Модельные переменные поддерживают автоматизацию сбора статистики.

Для хранения заявок, вошедших в систему, используется списковая структура issueList. Список имеет следующую структуру:

issueNum	isComplite[i]	priority	type	empId[i]	delay[i]

где:

i – номер стадии обслуживания:

- 0 – «Архитектура»;
- 1 – «Разработка»;

issueNum – номер заявки;

isComplite[i] – завершила ли заявка обслуживание на i-й стадии;

priority – идентификатор приоритета заявки;

type – идентификатор типа заявки;

empId[i] – идентификатор исполнителя, который занимался обслуживанием заявки на i-й стадии;

delay[i] – время (реализация случайной величины, закон распределения которой инкапсулирован в объект empDelay), которое должна обслуживаться заявка на i-й стадии исполнителем с идентификатором empId[i].

Списковая структура упорядочена по номеру заявки (issueNum).

Для хранения заявок, представляющих из себя очередь для каждого исполнителя на стадии «Архитектура», используется списковая структура QueueInArch*j*, где *j* – идентификатор исполнителя. Данная списковая структура создается для каждого исполнителя. Списки имеют следующую структуру:

issueNum

где:

issueNum – номер заявки.

Очереди упорядочены следующим образом: заявки с наибольшим приоритетом обслуживаются раньше, если у заявок одинаковый приоритет, то раньше обслуживается та, которая раньше попал в очередь.

Приоритет заявки определяется обращением к списковой структуре issueList. Поскольку issueList упорядочен по номеру заявки, то номер записи в списке однозначно соответствует номеру хранящейся в этой записи заявке. Таким образом, обращение к нужной записи списка issueList происходит по номеру issueNum, хранящемуся в очереди.

Для хранения заявок, представляющих из себя очередь для каждого исполнителя на стадии «Разработка», используется списковая структура QueueInDev*j*, где *j* – идентификатор исполнителя. Данная списковая структура создается для каждого исполнителя. Списки имеют следующую структуру:

issueNum

где:

issueNum – номер заявки.

Очереди упорядочены следующим образом: заявки с наибольшим приоритетом обслуживаются раньше, если у заявок одинаковый приоритет, то раньше обслуживается та, которая раньше попал в очередь.

Приоритет заявки определяется аналогичным образом как и для случая очередей исполнителей на стадии «Архитектура».

Как сказано ранее, ИМ строится с использованием СИМ Modelling [97]. В СИМ Modelling имеются встроенные модельные переменные. Текущее модельное время – одна из таких переменных, активно используемая в ИП.

Для определения текущего дня в модели используется следующая переменная:

currentDay – текущий день.

Для подсчета заявок, вошедших в систему, обслужившихся и заявок, обслуживающихся больше критического времени используются следующие модельные переменные:

issueCount – количество вошедших заявок;

issueCountByType[i] – количество вошедших заявок i–го типа;

issueCountByPriority[i] – количество вошедших заявок i–го приоритета;

issueCountServed – количество обслуженных заявок;

issueCountServedByType[i] – количество обслуженных заявок i–го типа;

issueCountServedByPriority[i] – количество обслуженных заявок i–го приоритета;

issueCountMoreCritical – количество заявок, пребывающих в системе более критического времени.

Для хранения информации о стадии «Архитектура» используется 3 модельных переменных, в которых j – идентификатор исполнителя. Далее приведены данные модельные переменные:

isWorkInArch[j] – работает ли j-й исполнитель на стадии «Архитектура» в данный момент (1 – работает, 0 – не работает);

isBusyInArch[j] – занят ли j-й исполнитель обслуживанием заявки на стадии «Архитектура» в данный момент (1 – занят, 0 – не занят);

issueNumInArch[j] – номер заявки, находящейся на обслуживании у j-го исполнителя на стадии «Архитектура» в данный момент (если 0, то у исполнителя нет заявки на обслуживании).

Для хранения информации о стадии «Разработка» используется 3 модельных переменных, в которых j – идентификатор исполнителя. Далее приведены данные модельные переменные:

isWorkInDev[j] – работает ли j-й исполнитель на стадии «Разработка» в данный момент (1 – работает, 0 – не работает);

isBusyInDev[j] – занят ли j-й исполнитель обслуживанием заявки на стадии «Разработка» в данный момент (1 – занят, 0 – не занят);

issueNumInDev[j] – номер заявки, находящейся на обслуживании у j-го исполнителя на стадии «Разработка» в данный момент (если 0, то у исполнителя нет заявки на обслуживании).

4.10.9. Моделирование случайных событий, распределений и потоков

В следствии того, что входной поток заявок является нестационарным в течении дня, а его интенсивность в будни дни существенно отличается от интенсивности в выходные, принято решение не моделировать входной поток традиционным методом (с использованием закона распределения времени между входами заявок). В качестве входного потока заявок в СМО в

определенный день используется реальная реализация входного потока заявок в отдел разработки ПО за день. Данный алгоритм возможно применить в модели, поскольку не стоит задача оценки влияния интенсивности входного потока на показатели эффективности. Алгоритм приведен далее:

1. В начале каждых суток в ИМ определяется, являются ли начавшиеся сутки будним или выходным днем.
2. Далее случайным образом выбирается реальная реализация входного потока заявок за сутки в отдел разработки ПО. Причем в зависимости от начавшихся суток просматриваются реализации будних или выходных дней.
3. На вход СМО подаются заявки в соответствии с моментами входа заявок из выбранной реализации.

Выбор конкретной реализации реального входного потока за сутки (раздельно для выходных и будних) – является случайным событием. Причем все реализации представляют из себя полную группу событий. В п.2 алгоритма для выбора случайным образом реализации реального входного потока за сутки используется классический метод моделирования полной группы случайных событий. Для этого используется случайное распределение псевдослучайных чисел РПВ на интервале [0; 1]. Для получения реализации используется первый независимый поток базовых псевдослучайных чисел incomingFlow.

Назначение заявки определенного типа является полной группой событий. Вероятность назначения заявки определенного типа заранее известна. Для моделирования типа заявки используется классический метод моделирования полной группы случайных событий. Для этого используется случайное распределение псевдослучайных чисел РПВ на интервале [0; 1]. Для получения реализации используется второй независимый поток базовых псевдослучайных чисел issueTypeStream.

Назначение заявки определенного приоритета является полной группой событий. Вероятность назначения заявки определенного приоритета заранее известна. Для моделирования типа заявки используется классический метод моделирования полной группы случайных событий. Для этого используется случайное распределение псевдослучайных чисел РПВ на интервале [0; 1]. Для получения реализации используется второй независимый поток базовых псевдослучайных чисел issuePriorityStream.

Заявки могут покидать СМО после окончания обслуживания на стадии «Архитектура». Определена оценка вероятности, с которой происходит данное событие. Для моделирования случайного события используется распределение псевдослучайных чисел РПВ на интервале [0; 1]. Для получения реализации используется четвертый независимый поток базовых псевдослучайных чисел leaveSystemAfterArchStream.

На стадии «Архитектура» в зависимости от исполнителя и типа, заявки могут обслуживаться случайное время. Оценка закона распределения времени обслуживания заявки на стадии «Архитектура» в зависимости от исполнителя и типа известна. По методу обратной функции происходит моделирование времени обслуживания заявки. Для этой цели используется распределение псевдослучайных чисел РПВ на интервале [0; 1]. Для получения реализации используется начиная с пятого и до четырнадцатого независимые потоки базовых псевдослучайных чисел DelayInArchStream*i,* где *i* – идентификатор исполнителя.

На стадии «Разработка» в зависимости от исполнителя и типа, заявки могут обслуживаться случайное время. Оценка закона распределения времени обслуживания заявки на стадии «Разработка» в зависимости от исполнителя и типа известна. По методу обратной функции происходит моделирование времени обслуживания заявки. Для этой цели используется распределение псевдослучайных чисел РПВ на интервале [0; 1]. Для получения реализации используется начиная с пятнадцатого и до двадцать четвертого независимые потоки базовых псевдослучайных чисел DelayInDevStream*i,* где *i* – идентификатор исполнителя.

4.10.10. Сбор статистики и расчет критериев

При расчете критериев предполагается, что процесс функционирования СМО стационарный и эргодический. Расчет временных критериев производится путем осреднения по времени.

Для сбора статистики по времени обслуживания заявки в ИМ используется модельная переменная timeInSystemAll. Во время покидания заявкой системы этой переменной присваивается время обслуживания заявки в системе. Таким образом, во время моделирования переменной будет присвоено все времена обслуживания заявок в системе. Статистический объект, связанный с этой переменной, позволяет построить гистограмму времени обслуживания заявки в системе.

Для сбора статистики по времени обслуживания заявки в системе, в зависимости от приоритета, используется модельная переменная timeInSystemByPriorrity[j], где j – идентификатор приоритета. Во время покидания заявкой системы этой переменной с номером, соответствующему идентификатору приоритета обслуженной заявки, присваивается время обслуживания заявки в системе. Таким образом, во время моделирования переменной с определенным номером будет присвоено все времена обслуживания заявок в системе с одним и тем же приоритетом. Статистический объект, связанный с этой переменной, позволяет определить оценку МО времени обслуживания заявки в системе в зависимости от приоритета.

Для сбора статистики по времени обслуживания заявки в системе, в зависимости от типа, используется модельная переменная timeInSystemByType[j], где j – идентификатор типа. Во

время покидания заявкой системы этой переменной с номером, соответствующему идентификатору типа обслуженной заявки, присваивается время обслуживания заявки в системе. Таким образом, во время моделирования переменной с определенным номером будет присвоено все времена обслуживания заявок в системе с одним и тем же типом. Статистический объект, связанный с этой переменной, позволяет определить оценку МО времени обслуживания заявки в системе в зависимости от типа.

Критическое время задается как параметр модели. После покидания заявкой системы происходит расчет критерия по формуле:

ProbInSystemMoreCritical = issueCountServed / issueCountMoreCritical,

где

issueCountServed – количество обслуженных заявок;

issueCountMoreCritical – количество заявок, пребывающих в системе более критического времени.

Для сбора статистики по загруженности исполнителей на каждой стадии («Архитектура» и «Разработка») используются модельные переменные probLoadInArch[i] и probLoadInDev[i] соответственно, где i – идентификатор исполнителя. При завершении моделирования происходит расчет этих переменных.

probLoadInArch[i] вычисляется, как отношение средневзвешенного значения переменной isBusyInArch[i] к средневзвешенному значению переменной isWorkInArch[i].

probLoadInDev[i] вычисляется, как отношение средневзвешенного значения переменной isBusyInDev к средневзвешенному значению переменной isWorkInDev.

Для сбора статистики по числу заявок в системе на конец недели используется модельная переменная issuesAtFriday. По окончании пятницы происходит вычисление, сколько заявок осталось в системе. Вычисление происходит по следующей формуле:

issuesAtFriday = issueCount - issueCountServed,

где

issueCount – количество вошедших заявок;

issueCountServed – количество обслуженных заявок.

4.10.11. Проектирование базы данных для хранения статистики

Для сбора статистики применялись запросы к БД АСУ РПО. В этой же БД созданы таблицы для хранения некоторой собранной статистики. В созданных таблицах хранятся:

- однопараметрические эмпирические функции распределения времени обслуживания заявок исполнителями в зависимости от типа заявки и стадии обслуживания;

- многомерные эмпирические функции распределения интервалов времени между входами заявок в СМО.

ER-модель таблиц, предназначенных для хранения статистики, а так же таблицы АСУ ПРО, которые применялись при сборе статистики приведены на рис. 4.29. На рисунке голубой цвет фона таблицы означает, что эта таблица является фрагментом БД АСУ РПО, белый цвет фона таблица означает, что эта таблица создана для хранения статистики. Связи между таблицами показаны линиями, причем белый ромб означает, что данная таблица является первичной, а черные круг – таблица является дочерней (т.е. в таблице с черным кругом в качестве внешнего ключа выступает первичный ключ таблицы с белым ромбом). В таблице 4.9. приведено описание таблиц ER-модели.

Таблица smo_variant предназначена для хранения описания различных вариантов собранной статистики, например, статистику можно собирать на различных временных интервалах. В поле comment необходимо записывать параметры собранной статистики.

В результате сбора статистики исследователь получает большое количество эмпирических функций распределения. Для хранения их описания используется таблица smo_functionDescription. Одновременно, кроме полей id и variant, могут быть заполнены поля emp, type, stage или поле isHoliday. Если поле isHoliday равняется null, то в записи содержится описание однопараметрической эмпирической функции распределения, в противном случае – многопараметрической.

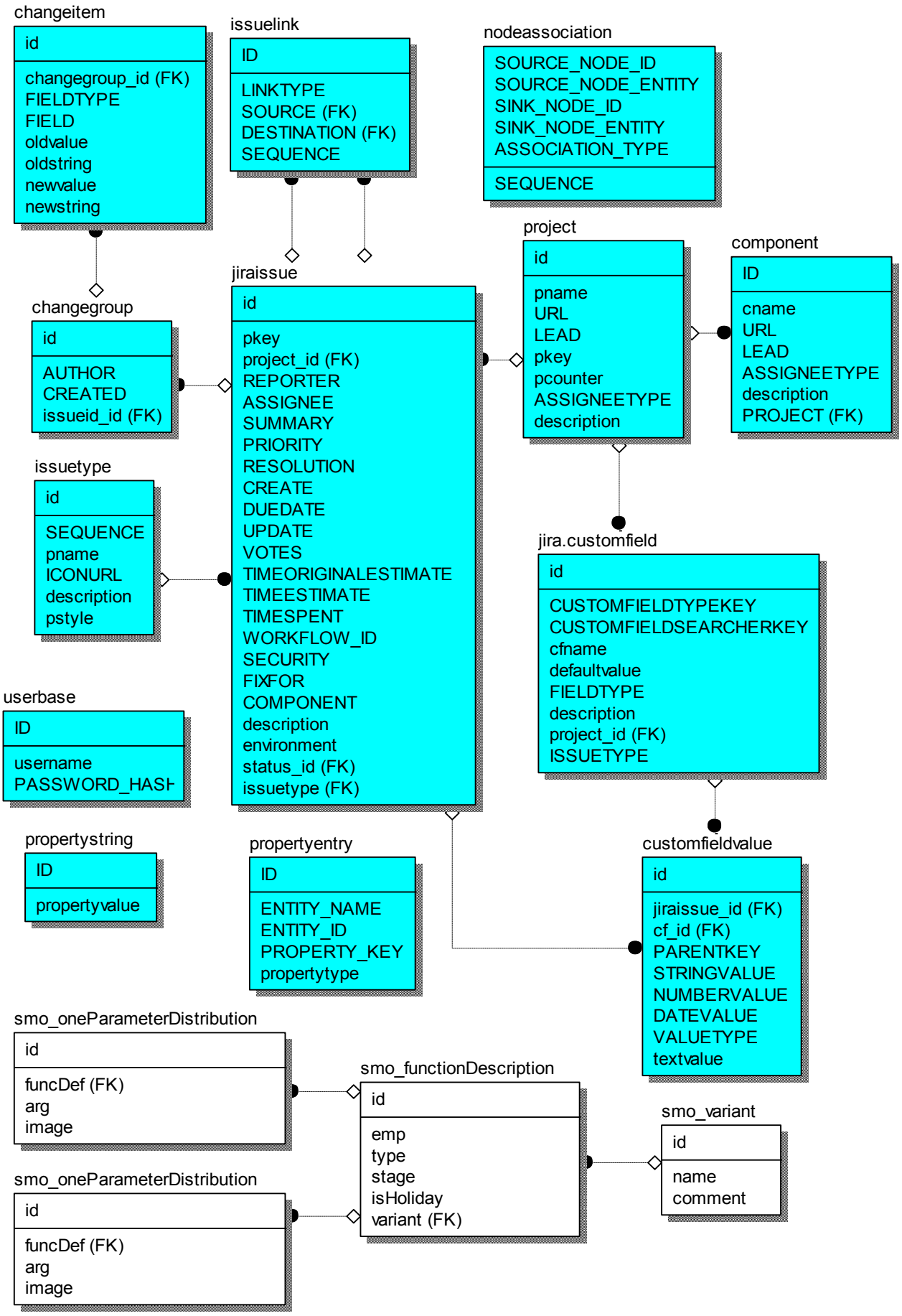


Рис. 4.29. ER-модель фрагмента БД АСУ ПРО, используемой для сбора статистики и таблицы, используемые для хранения собранной статистики.

Если поле isHoliday равняется 1, запись содержит описание многомерной эмпирической функции распределения в выходной день, если – 0, то будний.

Полное название однопараметрических эмпирических функций распределения – однопараметрические эмпирические функции распределения времени обслуживания заявок исполнителем в зависимости от типа заявки и стадии обслуживания. Итого, что бы однозначно определить однопараметрическую эмпирическую функцию необходимо задать: исполнителя, тип заявки и стадию обслуживания. Для этого служат поля emp, type и stage соответственно.

Таблицы smo_oneParameterDistribution и smo_multyParameterDistribution содержат значения функций «по точкам», т.е. в поле arg содержится аргумент функции, а в image – его изображение. При моделировании случайной величины, подчиненной однопараметрическому закону распределения, содержащемуся в таблице smo_oneParameterDistribution, моделирование происходит методом обратной функции, предварительно аппроксимировав эмпирическую функцию распределения кусочно-линейной функцией. Моделирование случайного вектора происходит методом обратной функции, но уже без аппроксимации.

Табл. 4.9.

Назначение таблиц БД.

Таблица	**Комментарий**
changegroup	Изменения заявок
changeitem	Детализация изменения заявок
component	Компоненты
customfieldvalue	Значение настраиваемого поля
issuelink	Связи заявок
issuetype	Типы заявок
customfield	Настраиваемые поля
jiraissue	Запросы
nodeassociation	Связи между сущностями
project	Проекты
propertyentry	Свойство
propertystring	Строковое свойство
userbase	Пользователи системы
smo_functionDescription	Описание эмпирических функций распределения
smo_multyParameterDistribution	Многопараметрические эмпирические функции распределения
smo_oneParameterDistribution	Однопараметрические эмпирические функции распределения
smo_variant	Вариант собранной статистики

4.10.12. Разработка программного обеспечения

Имитационная программа выполнена в среде программирования Delphi 7.0. в модульном виде [10, 153, 155, 168]. Это означает, что все функциональные единицы, используемые в программе, выполнены в виде отдельных модулей. На рис. 4.30. приведена схема структуры ПО

ИМ. Главный модуль Model.dpr создается автоматически средствами Delphi 7.0. и содержит в себе процедуру создания главного окна приложения и запускает цикл обработки сообщений ОС Windows.

Модуль UModel.pas занимает центральное место в ИП и содержит процедуры обработки модельных событий. Так же в начале и в конце этого модуля осуществляется обращение к модулю взаимодействия с внешними файлами для получения объектов, содержащих исходную статистическую информации и для записи результатов моделирования в выходной файл соответственно.

Модуль USettingsManager.pas предназначен для чтения и записи файла настроек модели. Под настройками понимаются имена входного и выходного файлов. При выходе из ИП в файл настроек записываются имена входного и выходного файлов, чтобы при следующем запуске ИП использовать их по умолчанию.

Модуль UInputManager.pas предназначен для чтения статистической информации из файла и формирования объектов, хранящих считанную информацию. Считанная статистическая информация инкапсулируется в объекты. Объекты, создаваемые модулем, в дальнейшем используются модулем UModel.pas для обработки модельных событий. Считываемый файл представляет из себя книгу Excel 2007.

Модуль UOutputManager.pas предназначен для записи результатов работы модели в файл. Модуль используется в UModel.pas в конце выполнения ИП для записи рассчитанных критериев. Выходной файл формируется на основании предварительно сформированного шаблона Excel 2007.

Модуль UEmp.pas содержит класс, объекты которого инкапсулируют перечень исполнителей с указанием времени, которое исполнитель работает на стадиях обслуживания. Класс модуля используется для создания одного объекта. Данные, которые содержит объект, хранятся на первой странице входного файла.

Модуль UPoligon.pas содержит класс, объекты которого инкапсулируют перечень типов заявок с частотой появления заявок определенного типа. Класс модуля используется для создания одного объекта. Данные, которые содержит объект, хранятся на второй странице входного файла.

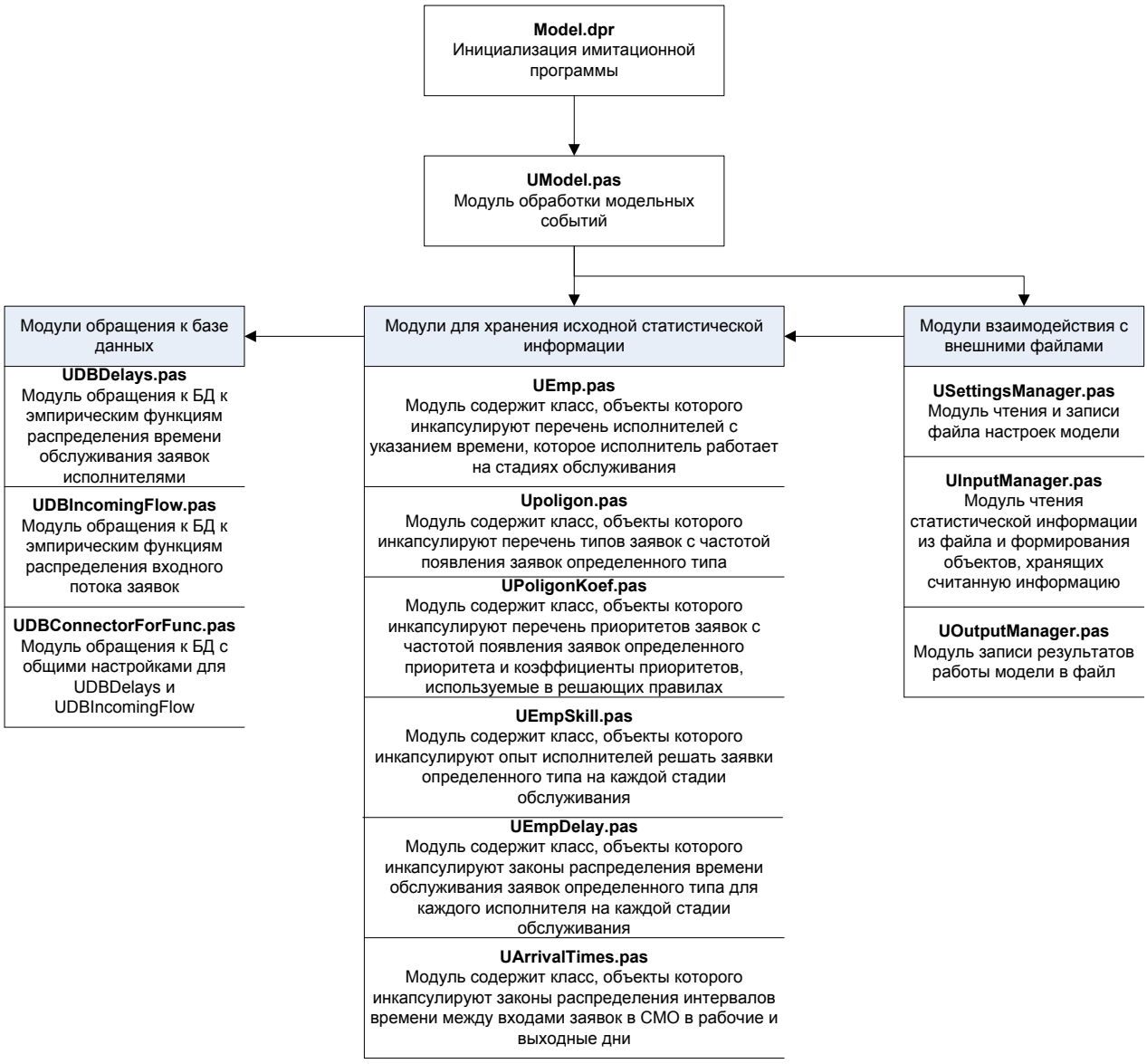


Рис. 4.30. Схема структуры ПО ИМ.

Модуль UPoligonKoef.pas содержит класс, объекты которого инкапсулируют перечень приоритетов заявок с частотой появления заявок определенного приоритета и коэффициенты приоритетов, используемые в решающих правилах. Класс модуля используется для создания одного объекта. Данные, которые содержит объект, хранятся на третьей странице входного файла.

Модуль UEmpSkill.pas содержит класс, объекты которого инкапсулируют опыт исполнителей решать заявки определенного типа на каждой стадии обслуживания. Класс модуля используется для создания одного объекта. Данные, которые содержит объект, хранятся на четвертой странице входного файла.

Модуль UEmpDelay.pas содержит класс, объекты которого инкапсулируют законы распределения времени обслуживания заявок определенного типа для каждого исполнителя на каждой стадии обслуживания. Класс модуля используется для создания одного объекта. Объект позволяет получать данные из БД (на основании эмпирической функции распределения) или

моделируя экспоненциальный закон распределения. Данные, которые содержит объект, хранятся на пятой странице входного файла.

Модуль UArrivalTimes.pas содержит класс, объекты которого инкапсулируют законы распределения интервалов времени между входами заявок в СМО в рабочие и выходные дни. Класс модуля используется для создания одного объекта. Объект получает данные, обращаясь к БД. Данные, которые содержит объект, хранятся на шестой странице входного файла.

Модуль UDBDelays.pas предназначен для обращения к эмпирическим функциям распределения времени обслуживания заявок исполнителями. Используется модулем UEmpDelay.pas.

Модуль UDBIncomingFlow.pas предназначен для обращения к эмпирическим функциям распределения входного потока заявок. Используется модулем UArrivalTimes.pas.

Модуль UDBConnectorForFunc.pas обращения к БД с общими настройками для UDBDelays.pas и UDBIncomingFlow.pas.

4.10.13. Алгоритмы обработки модельных событий

Центральное место в разработке имитационной программы занимают алгоритмы обработки модельных событий. В модели предусмотрено 7 модельных событий, алгоритмы которых с описанием приведены далее.

1-е модельное событие предназначено для генерации входных заявок на входе системы. Перед попаданием заявки в систему у нее устанавливается тип и приоритет. После входа заявки в систему выводится трассировка с описанием свершившегося события. Поскольку в модели моменты входа заявок в систему определяются заранее (в 7-м модельном событии), то в момент свершения 1-го модельного события известно, должны ли еще войти заявки в систему или нет. В случае, если заявки еще должны войти в систему, то планируется 1-е модельное событие. Далее, применяя решающее правило, определяется, какому исполнителю на стадии «Архитектура» заявка отправляется на обслуживание. Обслуживание заявки начинается в том случае, если в данный момент исполнитель работает на стадии «Архитектура» и он не занят другими заявками. В противном случае заявка встает в очередь этого исполнителя.

Если заявка все-таки попадает на обслуживание исполнителя, то планируется 2-е модельное событие. В завершении обработки события выводится трассировка со словом состояния.

Алгоритм обработки 1-го модельного события показан на рисунке 4.31.

2-е модельное событие предназначено для обработки события, связанного с окончанием обслуживания заявки на стадии «Архитектура», и определения дальнейших действий с обслуженной заявкой и освободившимся исполнителем. После окончания обслуживания заявки

выводится трассировка с описанием свершившегося события. Далее заявка удаляется с обслуживания и определяется, покидает ли она систему или ей требуется обслуживание на стадии «Разработка». В случае, если заявка покидает систему, производится расчет критериев оценки моделирования:

- оценка времени пребывания заявки в системе;
- оценка вероятности пребывания заявки в системе более критического времени.

В противном случае, применяя решающее правило, определяется, какому исполнителю на стадии «Разработка» заявка отправляется на обслуживание. Обслуживание заявки начинается в том случае, если в данный момент исполнитель работает на стадии «Разработка» и он не занято другими заявками. В противном случае заявка встает в очередь этого исполнителя.

Вне зависимости от описанных действий проверяется очередь освободившегося исполнителя на стадии «Архитектура». В случае, если его очередь не пуста, исполнитель берет следующую заявку, в соответствии с механизмом упорядочивания очередей и приступает к ее обслуживанию (планируется 2-е модельное событие). В противном случае, исполнитель простаивает. В завершении обработки события выводится трассировка со словом состояния.

Алгоритм обработки 2-го модельного события показан на рисунке 4.32.

3-е модельное событие предназначено для обработки события, связанного с окончанием обслуживания заявки на стадии «Разработка», и определения дальнейших действий с освободившимся исполнителем. После окончания обслуживания заявки выводится трассировка с описанием свершившегося события. Далее заявка удаляется с обслуживания и производится расчет критериев оценки моделирования:

- оценка времени пребывания заявки в системе;
- оценка вероятности пребывания заявки в системе более критического времени.

В заключении проверяется очередь освободившегося исполнителя на стадии «Разработка». В случае, если его очередь не пуста, исполнитель берет следующую заявку, в соответствии с механизмом упорядочивания очередей, и приступает к ее обслуживанию (планируется 3-е модельное событие). В противном случае, исполнитель простаивает. В завершении обработки события выводится трассировка со словом состояния.

Алгоритм обработки 3-го модельного события показан на рисунке 4.33.

4-е модельное событие предназначено для обработки события, связанного с началом рабочего дня и установкой всех исполнителей, работающих на стадии «Архитектура». В начале рабочего дня выводится трассировка с описанием свершившегося события. Далее организуется цикл по всем исполнителям. Внутри этого цикла текущий исполнитель назначается на работу на стадии «Архитектура». Далее проверяется, осталась ли у исполнителя недообслуженная заявка на стадии «Архитектура». В случае, если такая заявка имеется, исполнитель продолжает

ее обслуживать (помечается, как занятый на стадии «Архитектура»), в противном случае анализируется очередь исполнителя на стадии «Архитектура». В случае, если его очередь не пуста, исполнитель берет следующую заявку, в соответствии с механизмом упорядочивания очередей и приступает к ее обслуживанию (планируется 2-е модельное событие). В противном случае, исполнитель простаивает. В завершении цикла планируется 5-е модельное событие.

В завершении обработки события планируется 4-е модельное событие и выводится трассировка со словом состояния.

Алгоритм обработки 4-го модельного события показан на рисунке 4.34.

5-е модельное событие предназначено для обработки события, связанного с окончанием работы исполнителя на стадии «Архитектура» и, как следствие, началом работы на стадии «Разработка». В начале обработки события выводится трассировка с описанием свершившегося события. Далее исполнитель прекращает работку на стадии «Архитектура» (помечается, как не работающий и не занятый на стадии) и приступает к работе на стадии «Разработка». (помечается, как работающий на стадии). Далее проверяется, осталась ли у исполнителя недообслуженная заявка на стадии «Разработка». В случае, если такая заявка имеется, исполнитель продолжает ее обслуживать (помечается, как занятый на стадии «Разработка»), в противном случае анализируется очередь исполнителя на стадии «Разработка». В случае, если его очередь не пуста, исполнитель берет следующую заявку, в соответствии с механизмом упорядочивания очередей и приступает к ее обслуживанию (планируется 3-е модельное событие). В противном случае, исполнитель простаивает. В завершении обработки события планируется 6-е модельное событие и выводится трассировка со словом состояния.

Алгоритм обработки 5-го модельного события показан на рисунке 4.35.

6-е модельное событие предназначено для обработки события, связанного с окончанием работы исполнителя на стадии «Разработка» и, как следствие, окончанием рабочего дня исполнителя. В начале обработки события выводится трассировка с описанием свершившегося события. Далее исполнитель прекращает работку на стадии «Разработка» (помечается, как не работающий и не занятый на стадии). В завершении обработки события выводится трассировка со словом состояния.

Алгоритм обработки 6-го модельного события показан на рисунке 4.36.

7-е модельное событие предназначено для обработки события, связанного с окончанием суток (началом новых суток). В начале обработки события выводится трассировка с описанием свершившегося события. Далее проверяется, завершившейся день – пятница, если да, то собирается статистика по количеству заявок в системе на конец рабочей недели. Далее заполняется список с моментами входа заявок в начавшийся день на основании того, является ли начавшийся день выходным или будним. Если в начавшийся день должны войти заявки, то

планируется 1-е модельное событие. В завершении обработки события планируется 7-е модельное событие и выводится трассировка со словом состояния.

Алгоритм обработки 7-го модельного события показан на рисунке 4.37.

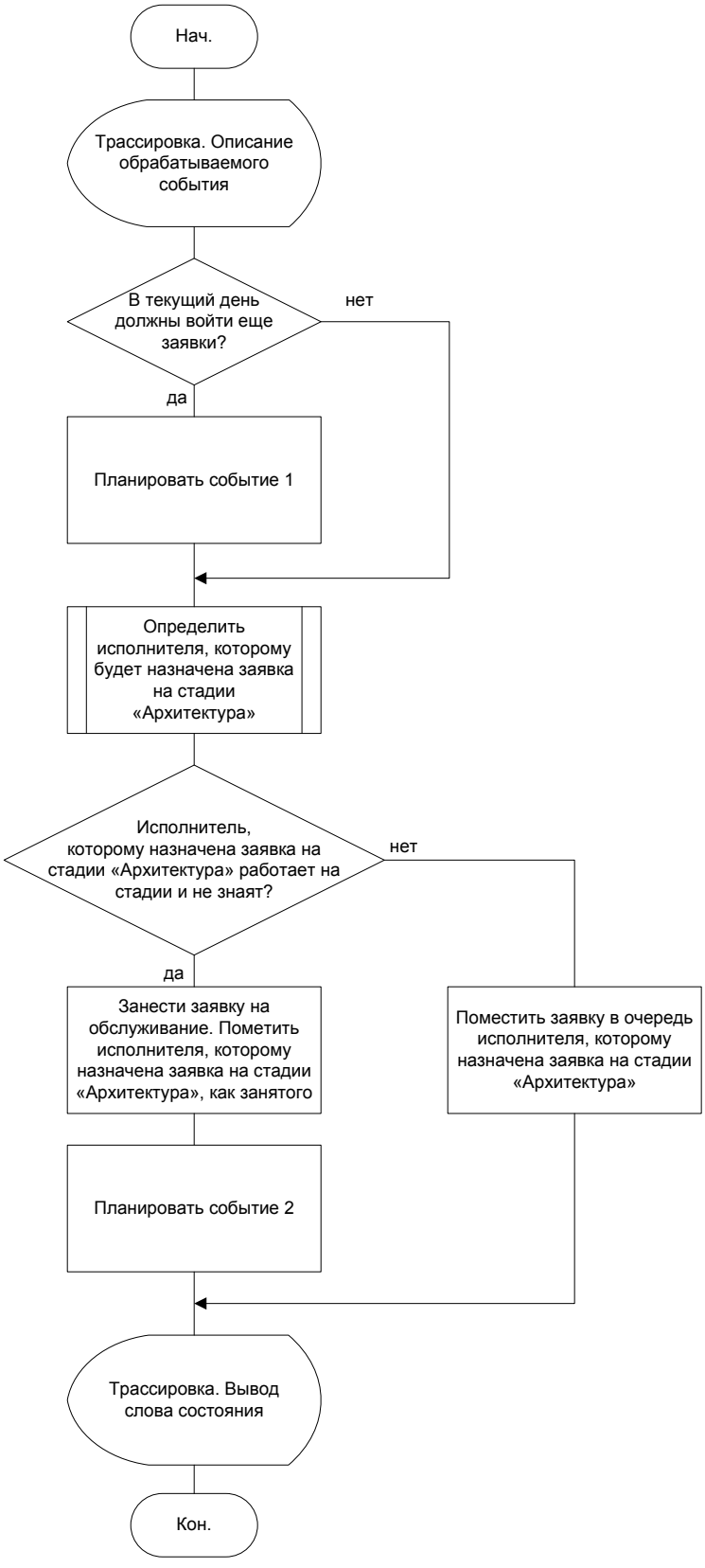


Рис. 4.31. Алгоритм обработки 1-го модельного события.

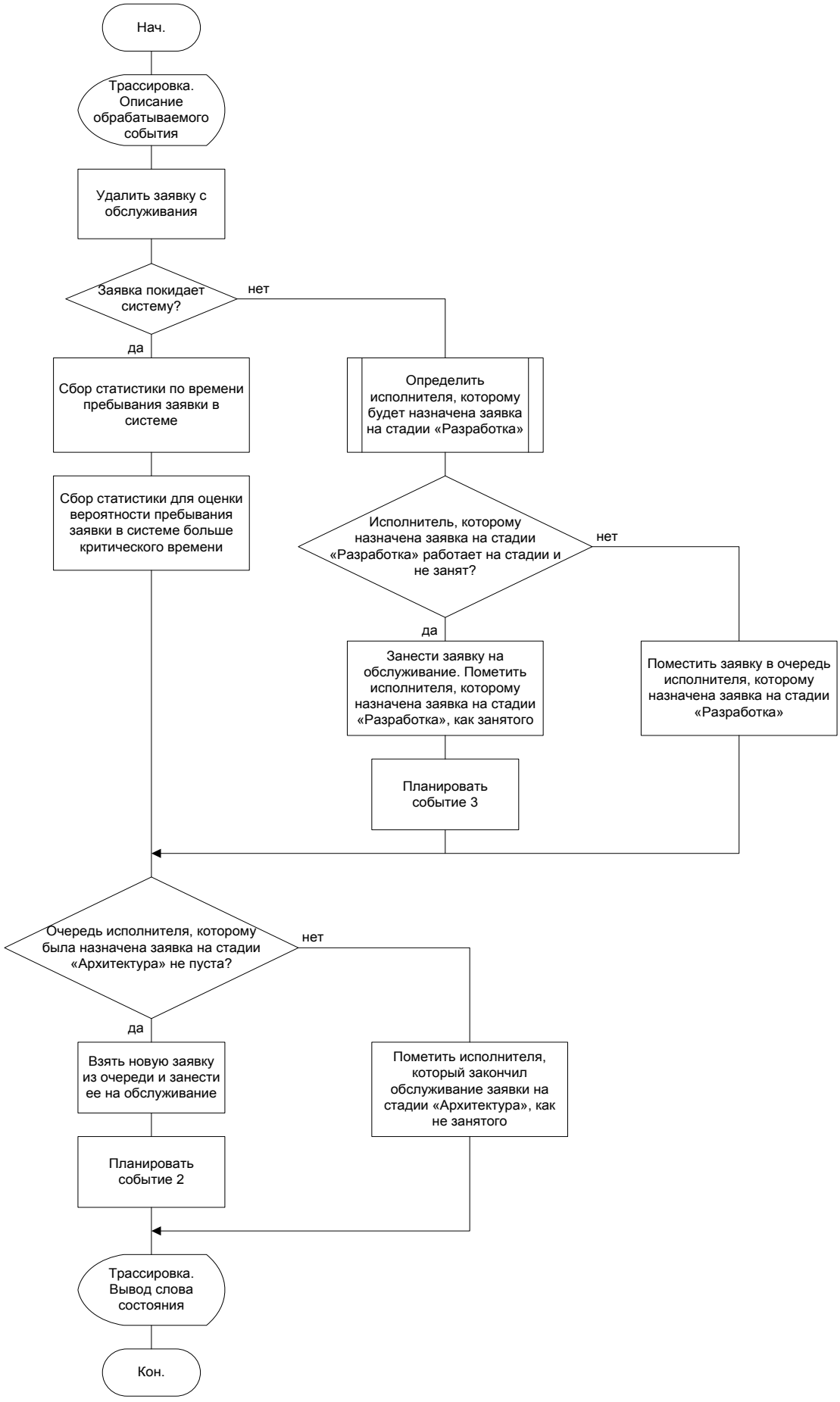


Рис. 4.32. Алгоритм обработки 2-го модельного события.

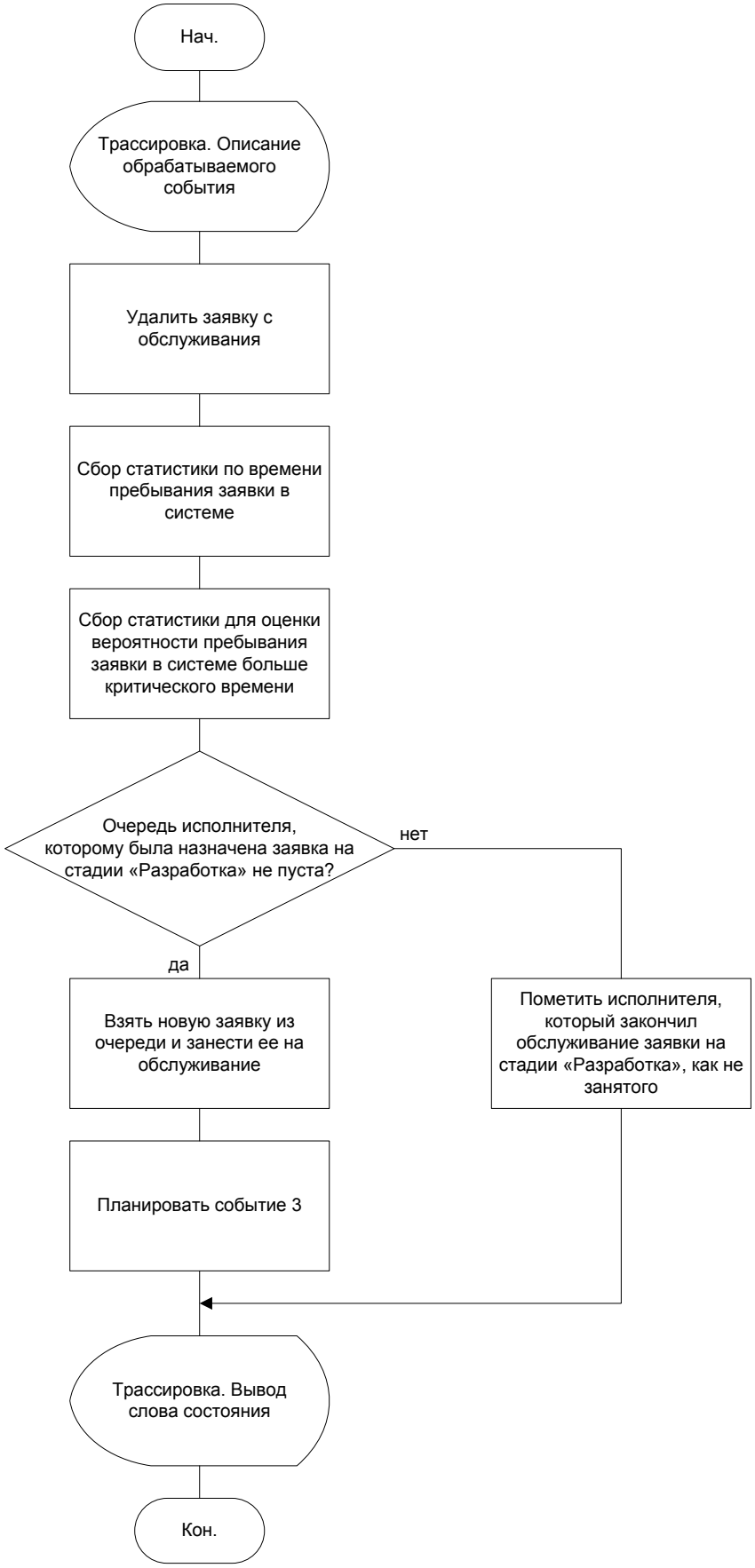


Рис. 4.33. Алгоритм обработки 3-го модельного события.

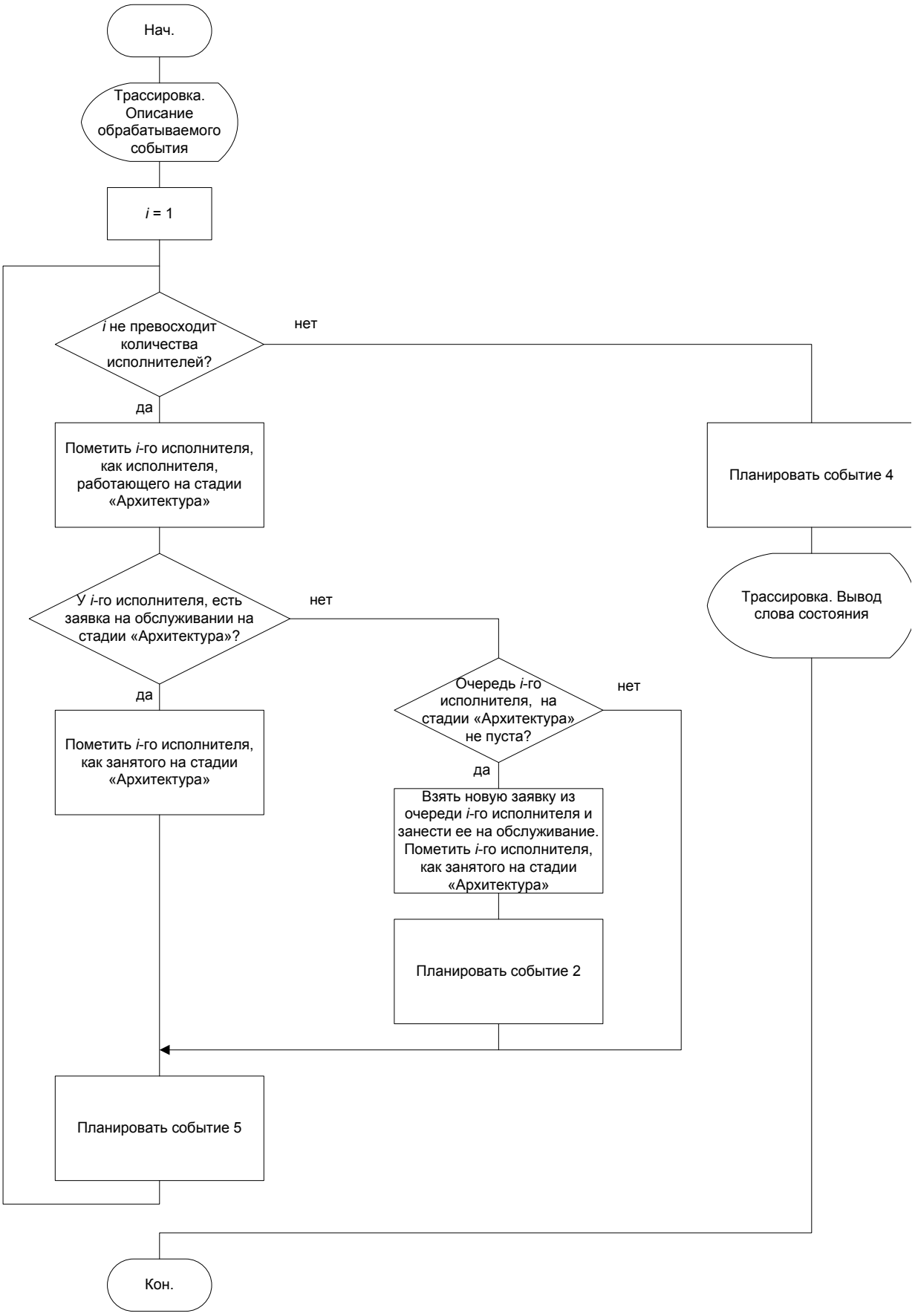


Рис. 4.34. Алгоритм обработки 4-го модельного события.

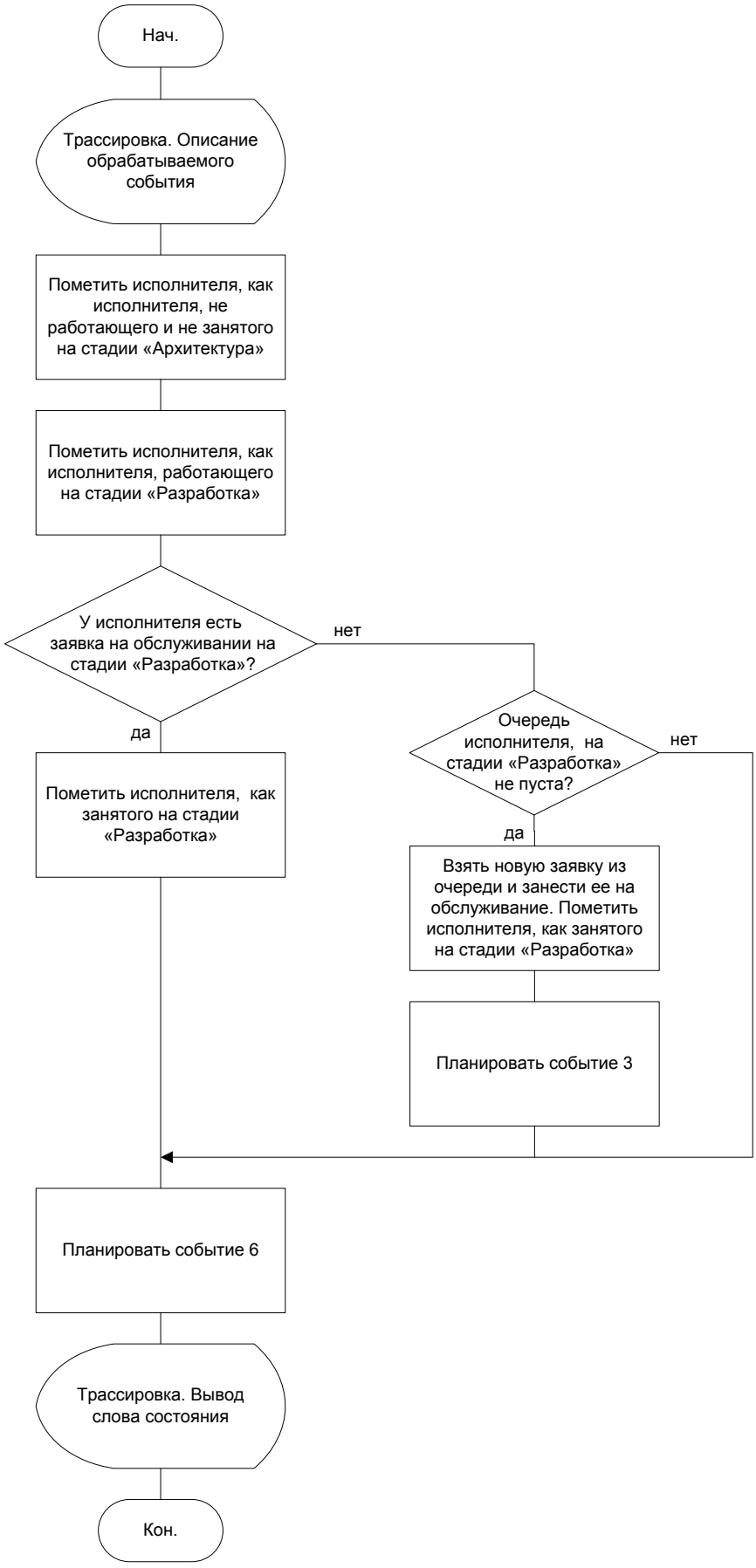


Рис. 4.35. Алгоритм обработки 5-го модельного события.

Рис. 4.36. Алгоритм обработки 6-го модельного события.

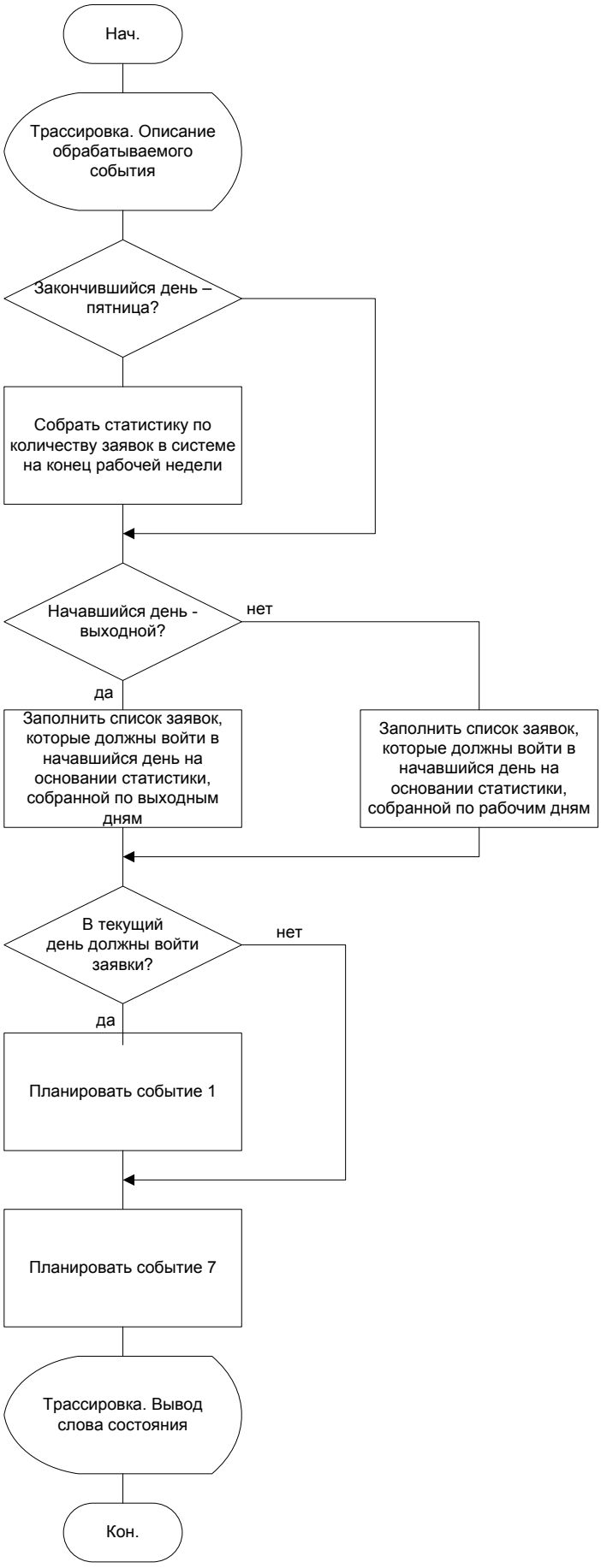


Рис. 4.37. Алгоритм обработки 7-го модельного события.

4.10.14. Имитационные эксперименты

Для отображения хода моделирования в ИП предусмотрена трассировка. Эта трассировка выводится во время прогона ИП в окно трассировки, а по окончанию моделирования записывается в файл trace.txt директории, из которой запускалась ИП. Для каждого модельного события трассировочная печать выводится в следующем формате:

- описание свершившегося модельного события;
- описания запланированных модельных событий;
- слово состояния.

Описание свершившегося модельного события состоит из следующих составляющих:

- заголовок модельного события;
- дополнительные атрибуты модельного события.

Заголовок модельного события состоит из следующих составляющих:

- время свершения модельного события;
- код модельного события.

Дополнительные атрибуты модельного события различаются между собой в зависимости от свершившегося события.

Для 1-го модельного события дополнительные атрибуты выводятся в следующем порядке:

- номер заявки;
- идентификатор типа заявки;
- идентификатор приоритета заявки.

Для 2 и 3-го модельного события дополнительные атрибуты выводятся в следующем порядке:

- идентификатор сотрудника;
- номер заявки.

Для 4-го модельного события дополнительные атрибуты не выводятся.

Для 5 и 6-го модельного события дополнительные атрибуты выводятся в следующем порядке:

- идентификатор сотрудника.

Для 7-го модельного события дополнительные атрибуты не выводятся.

При обработке свершившегося модельного события могут планироваться модельные события. Их описание выводится в порядке их занесения в СБС и имеет формат, идентичный описанию свершившегося модельного события.

Слово состояния отражает содержание модельных переменных и их производных, например, длины очередей. Слово состояния состоит из следующих составляющих:

- общие сведения:
- текущий день;
- количество заявок, вошедших в систему.

Исполнители на стадии «Архитектура». По каждому Исполнителю выводится следующая информация:

- работает ли исполнитель на стадии «Архитектура» (0 – не работает, 1 – работает);
- занят ли исполнитель обслуживанием заявки на стадии «Архитектура» (0 – не работает, 1 – работает);
- номер обслуживаемой исполнителем заявки;
- длина очереди у исполнителя.

Исполнители на стадии «Разработка». По каждому исполнителю выводится следующая информация:

- работает ли исполнитель на стадии «Разработка» (0 – не работает, 1 – работает);
- занят ли исполнитель обслуживанием заявки на стадии «Разработка» (0 – не работает, 1 – работает);
- номер обслуживаемой исполнителем заявки;
- длина очереди исполнителя.

В рамках одной стадии сведения об исполнителях отделяются друг от друга квадратными скобками. Стадии отделяются друг от друга фигурными скобками.

Далее приведен пример трассировочной печати для двух исполнителей:

10.984| Event1| 2| 1| 1|

13.205| Event1| 3| 4| 1|

11.019| Event2| 0| 2|

1| 2{[1|1|2|0][1|0|0|0]} {[0|0|0|0][0|0|0|0]}

В первой строке содержится описание свершившегося модельного события. 10.984 – модельное время свершения события с кодом Event1 (1-е модельное событие). Событие свершилось для заявки с номером 2, идентификатором типа 1 и приоритета 1. Все идентификаторы начинают нумерацию с 0, а заявки с 1.

В результате свершения 1-го модельного события произошло планирование 2-х модельных событий. Первое событие – событие с кодом Event1 (1-е модельное событие) для заявки с номером 3, идентификатором типа 4 и приоритета 1. Второе событие – событие с кодом Event2 (2-е модельное событие) для сотрудника с идентификатором 0 и заявки номер 2.

Последняя строка содержит описание модельных переменных: текущий день – 1, в систему вошло 2 заявки. На стадии «Архитектура»:

0-й сотрудник -

- работает на стадии;
- занят обслуживанием запроса;
- обслуживает запрос с номером 2;
- очереди у сотрудника нет.

1-й сотрудник -

- работает на стадии;
- не занят обслуживанием запроса;
- нет запроса на обслуживании;
- очереди у сотрудника нет.

На стадии «Разработка»:

0-й сотрудник -

- не работает на стадии;
- не занят обслуживанием запроса;
- нет запроса на обслуживании;
- очереди у сотрудника нет.

1-й сотрудник -

- не работает на стадии;
- не занят обслуживанием запроса;
- нет запроса на обслуживании;
- очереди у сотрудника нет.

Словесно сложившуюся ситуацию можно описать так: в момент времени 10.984 в систему вошла вторая заявка. Решающим правилом была назначена исполнителю с идентификатором 0. Т.к. он работал в этот момент на стадии «Архитектура» и не был занят никакими запросами, он преступил к ее обслуживанию. Обслуживание заявки сотрудника завершит в 11.019. следующая заявка войдет в систему в 13.205.

Проверка работы ИМ.

Проверка работы ИМ основана на анализе фрагментов трассировочной печати, охватывающей слово состояния предыдущего модельного события и полное описание свершившегося модельного события (описание свершившегося модельного события, описания запланированных модельных событий, слово состояния). Анализ заключается в проверке изменения слова состояния на произошедшее модельное событие.

Анализ строится следующим образом:

1. Рассматривается свершившееся модельное событие.
2. Рассматриваются запланированные модельные события.
3. В соответствии со свершившемся (запланированными) модельным событием проверяется, правильно ли изменились модельные переменные путем сравнения слова состояния предыдущего модельного события и свершившегося.

Рассмотрим следующий листинг трассировочной печати:

```
0.000| Event7|
9.000| Event4|
 0|  0{[0|0|0|0][0|0|0|0]} {[0|0|0|0][0|0|0|0]}

 0.000| Event7|
10.023| Event1|   1|   1|   1|
24.000| Event7|
 1|  0{[0|0|0|0][0|0|0|0]} {[0|0|0|0][0|0|0|0]}

 9.000| Event4|
13.759| Event5|   0|
13.138| Event5|   1|
33.000| Event4|
 1|  0{[1|0|0|0][1|0|0|0]} {[0|0|0|0][0|0|0|0]}

10.023| Event1|   1|   1|   1|
10.984| Event1|   2|   1|   1|
10.932| Event2|   0|   1|
 1|  1{[1|1|1|0][1|0|0|0]} {[0|0|0|0][0|0|0|0]}

10.932| Event2|   0|   1|
 1|  1{[1|0|0|0][1|0|0|0]} {[0|0|0|0][0|0|0|0]}

10.984| Event1|   2|   1|   1|
13.205| Event1|   3|   4|   1|
11.019| Event2|   0|   2|
 1|  2{[1|1|2|0][1|0|0|0]} {[0|0|0|0][0|0|0|0]}
```

11.019| Event2| 0| 2|
1| 2{[1|**0**|**0**|0][**1**|0|0|0]} {[0|0|0|0][**0**|0|0|0]}

13.138| Event5| 1|
17.000| Event6| 1|
1| **2**{[1|0|0|0][**0**|0|0|**0**]} {[0|0|0|0][**1**|0|0|0]}

13.205| Event1| 3| 4| 1|
13.468| Event1| 4| 1| 2|
1| **3**{[1|0|0|0][0|0|0|**1**]} {[0|0|0|0][1|0|0|0]}

Рассмотрим 1-е модельное событие. В момент времени 10.023 свершилось 1-е модельное событие (вход заявки в систему) и запланировано следующее 1-е модельное событие и 2-е модельное событие (окончание обслуживания заявки на стадии «Архитектура»). Количество заявок в системе должно увеличиться на 1. Это отражает изменение составляющей слова состояния: «Количество заявок, вошедших в систему» с 0 на 1. После входа заявки в систему она должна быть назначена определенному исполнителю на стадии «Архитектура». В данном случае она назначена исполнителю с идентификатором 0 и встала на обслуживание, т.к. в данный момент исполнитель работает на стадии «Архитектура». Это отражает изменение составляющих слова состояния: «Номер обслуживаемой сотрудником заявки» с 0 на 1 (т.к. в систему вошла заявка с номером 1); «Занят ли исполнитель обслуживанием заявки на стадии «Архитектура» с 0 на 1.

Рассмотрим еще одно 1-е модельное событие. В момент времени 13.205 свершилось 1-е модельное событие (вход заявки в систему) и запланировано следующее 1-е модельное событие. Количество заявок в системе должно увеличиться на 1. Это отражает изменение составляющей слова состояния: «Количество заявок, вошедших в систему» с 2 на 3. После входа заявки в систему она должна быть назначена определенному исполнителю на стадии «Архитектура». В данном случае она назначена исполнителю с идентификатором 1 и встала ему в очередь, т.к. в данный момент исполнитель не работает на стадии «Архитектура». Это отражает изменение составляющей слова состояния: «Длина очереди у исполнителя» с 0 на 1.

Рассмотрим 2-е модельное событие. В момент времени 11.019 свершилось 2-е модельное событие (окончание обслуживания заявки на стадии «Архитектура»), но событий не запланировано. Это связано с тем, что у сотрудника с идентификатором 0 нет заявок в очереди и, как следствие, он не может себе поставить запрос на обслуживание. Изменились следующие составляющие слова состояния: «Занят ли исполнитель обслуживанием заявки на стадии

«Архитектура» с 1 на 0; «Номер обслуживаемой исполнителем заявки» с 2 на 0 (т.к. закончилось обслуживание 2-й заявки).

Рассмотрим 4-е модельное событие. В момент времени 9.000 свершилось 4-е модельное событие (начало работы исполнителей на стадии «Архитектура»). В результате его свершения запланировано два 5-х модельных события, по одному на каждого исполнителя (исполнитель окончил работать на стадии «Архитектура», начал работать на стадии «Разработка») и 4-е модельное событие. Все исполнители начали работать на стадии «Архитектура», т.к. у них очереди пусты и на обслуживании у них не было заявок, то они просто начали работать на стадии «Архитектура». Это отражает изменение составляющей слова состояния: «Работает ли исполнитель на стадии «Архитектура» с 0 на 1.

Рассмотрим 5-е модельное событие, которое было запланировано в результате ранее рассмотренного 4-го модельного события. В момент времени 13.138 Свершилось 5-е модельное событие (исполнитель окончил работать на стадии «Архитектура», начал работать на стадии «Разработка») и запланировано 6-е модельное событие (исполнитель окончил работать на стадии «Разработка»). Исполнитель с идентификатором 1 окончил работать на стадии «Архитектура» и начал работать на стадии «Разработка». Ни в «Архитектуре», ни в «Разработке» у исполнителя нет заявок, поэтому он просто перешел на другую стадию. Это отражает изменение составляющей слова состояния стадии «Архитектура»: «Работает ли исполнитель на стадии «Архитектура» с 1 на 0. И изменение составляющей слова состояния стадии «Разработка»: «Работает ли в данный момент сотрудник на стадии «Разработка» с 0 на 1.

Рассмотрим 7-е модельное событие. В момент времени 0,000 свершилось 7-е модельное событие (конец очередных суток) и запланированы 1-е модельное событие (вход заявки в систему) и 7-е модельное событие (конец очередных суток). Т.к. начался новый день, то в модели планируется новая реализация входного потока заявок на текущий день. Начался новый день с номером 1. Планируется окончание очередного рабочего дня, которое должно произойти в 1 * 24 = 24. Это отражает изменение составляющей слова состояния: «Текущий день» с 0 на 1.

Рассмотрим следующий листинг трассировочной печати:

4| 26{[**1**|**1**|4|7][0|0|21|6]} {[**0**|**0**|16|0][1|0|0|0]}

85.759| Event5| 0|

89.000| Event6| 0|

4| 26{[**0**|**0**|4|7][0|0|21|6]} {[**1**|**1**|**16**|0][1|0|0|0]}

86.483| Event3| 0| 16|

4| 26{[0|0|4|7][0|0|21|6]} {[1|**0**|**0**|0][1|0|0|0]}

89.000| Event6| 0|
4| 26{[0|0|4|7][0|0|21|6]} {[0|0|0|0][**1**|0|0|0]}

89.000| Event6| 1|
4| 26{[0|0|4|7][0|0|21|6]} {[0|0|0|0][**0**|0|0|0]}

96.000| Event7|
106.038| Event1| 27| 4| 2|
120.000| Event7|
5| 26{[**0**|**0**|4|7][**0**|**0**|21|6]} {[0|0|0|0][0|0|0|0]}

105.000| Event4|
109.759| Event5| 0|
109.138| Event5| 1|
177.000| Event4|
5| 26{[**1**|**1**|4|7][**1**|**1**|21|6]} {[0|0|0|0][0|0|0|0]}

105.618| Event2| 1| 21|
107.218| Event2| 1| 22|
5| 26{[1|1|4|7][1|1|22|5]} {[0|0|0|0][0|0|0|0]}

106.038| Event1| 27| 4| 2|
106.782| Event1| 28| 4| 2|
5| 27{[1|1|4|7][1|1|22|6]} {[0|0|0|0][0|0|0|0]}

Рассмотрим 5-е модельное событие в момент времени 85.759. Свершилось 5-е модельное событие (исполнитель окончил работать на стадии «Архитектура», начал работать на стадии «Разработка») и запланировано 6-е модельное событие (исполнитель окончил работать на стадии «Разработка»). Исполнитель с идентификатором 0 окончил работать на стадии «Архитектура» и начал работать на стадии «Разработка». В «Архитектуре» и в «Разработке» у исполнителя были заявки на обслуживании, поэтому он прервал обслуживания заявки на стадии «Архитектура» и приступил к обслуживанию другой заявки на стадии «Разработка». Это отражает изменение составляющих слова состояния стадии «Архитектура»: «Работает ли исполнитель на стадии «Архитектура» с 1 на 0; «Занят ли исполнитель обслуживанием заявки на стадии «Архитектура» с 1 на 0. И изменение составляющих слова состояния стадии

«Разработка»: «Работает ли в данный момент сотрудник на стадии «Разработка» с 0 на 1; «Занят ли исполнитель обслуживанием заявки на стадии «Разработка» с 0 на 1.

Рассмотрим 3-е модельное событие. В момент времени 86.489 свершилось 3-е модельное событие (окончание обслуживания заявки на стадии «Разработка»). Заявка должна покинуть стадию «Разработка» и систему. Очереди у исполнителя на стадии «Разработка» нет, поэтому он не может взять следующую заявку на обслуживание. Это отражает изменение составляющих слова состояния: «Занят ли исполнитель обслуживанием заявки на стадии «Архитектура» с 1 на 0; «Номер обслуживаемой исполнителем заявки» с 16 на 0.

Рассмотрим 6-е модельное событие. В момент времени 89.000 свершилось два 6-х модельных события (окончание работы исполнителя на стадии «Разработка»). Рассмотрим второе из них. Исполнитель с идентификатором 1 закончил работать на стадии «Разработка». Т.к. у него не было заявок на обслуживании, то он просто прекратил работать на стадии «Разработка». Это отражает изменение составляющей слова состояния: «Работает ли исполнитель на стадии «Разработка» с 1 на 0.

Рассмотрим 4-е модельное событие. В момент времени 105.000 свершилось 4-е модельное событие (начало работы исполнителей на стадии «Архитектура»). В результате его свершения запланировано два 5-х модельных события, по одному на каждого исполнителя (исполнитель окончил работать на стадии «Архитектура», начал работать на стадии «Разработка») и 4-е модельное событие. Все исполнители начали работать на стадии «Архитектура», т.к. у них на обслуживании были заявки, то они продолжают их обслуживать. Это отражает изменение составляющих слова состояния: «Работает ли исполнитель на стадии «Архитектура» с 0 на 1; «Занят ли исполнитель обслуживанием заявки на стадии «Архитектура» с 0 на 1.

Рассмотрим следующий листинг трассировочной печати:

6| 41{[0|0|4|13][0|0|25|10]} {[0|0|23|0][0|0|0|0]}

144.000| Event7|
168.000| Event7|
7| 41{[0|0|4|13][0|0|25|10]} {[0|0|23|0][0|0|0|0]}

168.000| Event7|
182.573| Event1| 42| 1| 2|
192.000| Event7|
8| 41{[0|0|4|13][0|0|25|10]} {[0|0|23|0][0|0|0|0]}

177.000| Event4|

181.759| Event5| 0|

181.138| Event5| 1|

201.000| Event4|

8| 41{[1|1|4|13][1|1|**25**|**10**]} {[0|0|23|0][0|0|0|0]}

178.521| Event2| 1| 25|

202.849| Event2| 1| 30|

8| 41{[1|1|4|13][1|1|**30**|**9**]} {[0|0|23|0][0|0|0|**1**]}

181.138| Event5| 1|

185.000| Event6| 1|

8| 41{[1|1|4|13][0|0|30|9]} {[0|0|23|0][1|0|0|0]}

Рассмотрим 7-е модельное событие. В момент времени 144.000 свершилось 7-е модельное событие (конец очередных суток) и запланировано 7-е модельное событие (конец очередных суток). 1-е модельное событие (вход заявки в систему) не запланировано, т.к. начавшийся день – выходной, а в выходные вероятность входа заявок существенно уменьшается.

Рассмотрим 2-е модельное событие. В момент времени 178.521 свершилось 2-е модельное событие (окончание обслуживания заявки на стадии «Архитектура») и запланировано 2-е модельное событие. Это связано с тем, что у исполнителя с идентификатором 1 в очереди находилось 10 заявок и он взял 30-ю на обслуживание. После стадии «Архитектура» заявка попала на стадию «Разработка» к исполнителю с идентификатором 1 и встала к нему в очередь. Изменились следующие составляющие слова состояния стадии «Архитектура»: «Номер обслуживаемой исполнителем заявки» с 25 на 30; «Длина очереди у исполнителя» с 10 на 9. Так же изменилась составляющая слова состояния стадии «Разработка»: «Длина очереди у исполнителя» с 0 на 1.

На приведенных листингах рассмотрены все семь модельных событий. Некоторые модельные события рассматривались несколько раз, что бы проверить изменение слова состояния в различных условиях. Изменения слова состояния соответствует заложенному в имитационную программу алгоритму и, следовательно, соответствует логике функционирования системы.

4.10.15. Верификация модели

Суть верификации модели заключается в варьировании параметров системы и последующим анализом критериев оценки моделирования системы. На каждом этапе верификации модели необходимо менять лишь один параметр модели, что бы избежать влияния на критерии оценки моделирования других параметров системы. При изменении параметров заведомо известно как повлияет это на характер изменения критериев оценки моделирования (увеличат эти оценки или уменьшат). Верификация модели состоит в том, чтобы убедиться, что модель изменяет критерии оценки моделирования в ответ на варьирования параметров в соответствии с заведомо известным характером изменения этих критериев.

Варьирование количества исполнителей.

При увеличении количества исполнителе значения всех критерии: оценка МО времени обслуживании заявок в системе, вероятность обслуживать заявку более критического времени, загрузка исполнителей и оценка МО числа заявок в системе на конец недели – должны уменьшиться. И, наоборот, при уменьшении количества исполнителей все значения перечисленных критериев должны увеличиться (см. табл. 4.10 и рис. 4.38-4.41). При изменении количества исполнителей их опыт так же должен меняться, поскольку величина опыта зависит от количества исполнителей. Опыт удаленного исполнителя обслуживать заявки определенного типа равномерно распределяется между ненулевым опытом других исполнителей. Время моделирования составляет 180 суток. Критическое время составляет 30 суток. Остальные параметры взяты, как в собранной статистике.

Табл. 4.10.

Изменение критериев оценки моделирования
в результате варьирования количества исполнителей

Критерий	**Количество исполнителей**			
	2	**3**	**4**	**5**
Оценка МО времени обслуживании заявок в системе	485,205	255,003	193,850	98,326
Вероятность обслуживать заявку более критического времени	0,305	0,097	0,061	0,015
Средняя загруженность всех исполнителей на стадии «Архитектура»	0,991	0,943	0,798	0,770
Средняя загруженность всех исполнителей на стадии «Разработка»	0,992	0,982	0,967	0,759
Оценка МО числа заявок в системе на конец недели	210,153	98,115	56,769	19,154

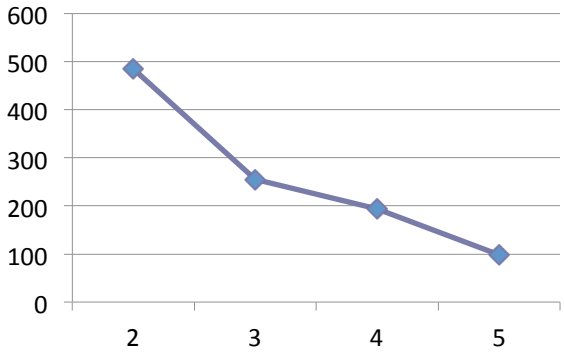


Рис. 4.38. Изменение оценки МО времени обслуживании заявок в системе при изменении количества исполнителей.

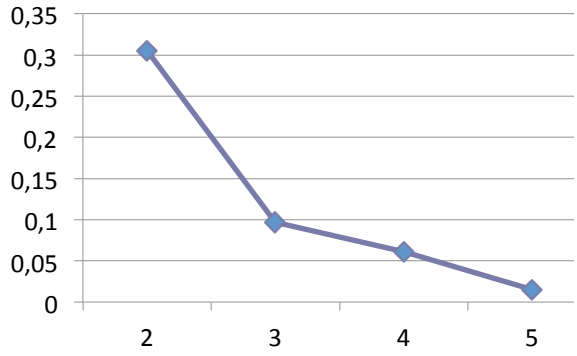


Рис. 4.40. Изменение вероятности обслуживать заявку более критического времени при изменении количества исполнителей.

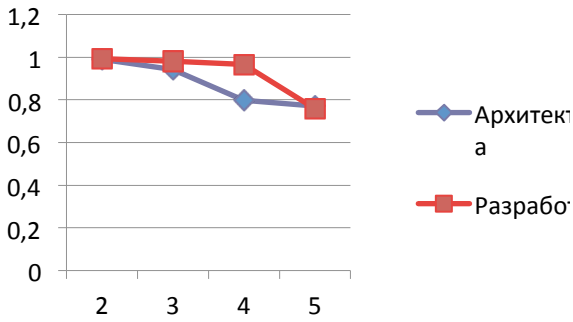


Рис. 4.39. Изменение средней загруженности всех исполнителей на обоих стадиях обслуживания при изменении количества исполнителей.

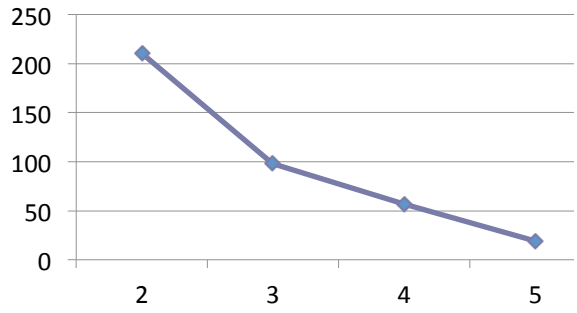


Рис. 4.41. Изменение оценки МО числа заявок в системе на конец недели при изменении количества исполнителей.

Варьирование вероятности покинуть заявкой систему после стадии «Архитектура».

При увеличении вероятности покинуть заявкой систему после стадии «Архитектура» критерии: оценка МО времени обслуживании заявок в системе, вероятность обслуживать заявку более критического времени, загрузка исполнителей на стадии «Разработка» и оценка МО числа заявок в системе на конец недели – должны уменьшиться. И наоборот, при уменьшении вероятности покинуть систему после стадии «Архитектура» все значения перечисленных критериев должны увеличиться (см. табл. 4.11 и рис. 4.42-4.45). Критерий загрузка исполнителей на стадии «Архитектура» не должен изменяться при изменении варьируемой вероятности. Время моделирования составляет 180 суток. Критическое время составляет 30 суток.

Табл. 4.11.
Изменение критериев оценки моделирования в результате варьирования вероятности покинуть заявкой систему после стадии «Архитектура».

Критерий	Вероятность покинуть систему после стадии «Архитектура»				
	0	0,25	0,5	0,75	1
Оценка МО времени обслуживании заявок в системе	433,884	177,262	120,824	75,552	63,244
Вероятность обслуживаться заявку более критического времени	0,147	0,048	0,011	0,009	0,008
Средняя загруженность всех исполнителей на стадии «Архитектура»	0,763	0,763	0,763	0,763	0,763
Средняя загруженность всех исполнителей на стадии «Разработка»	0,971	0,945	0,871	0,455	0
Оценка МО числа заявок в системе на конец недели	109,077	47,692	26,923	16,846	14,462

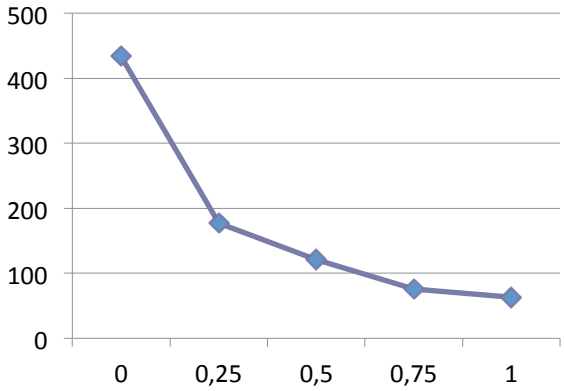


Рис. 4.42. Изменение оценки МО времени обслуживании заявок в системе при изменении вероятности покинуть заявкой систему после стадии «Архитектура».

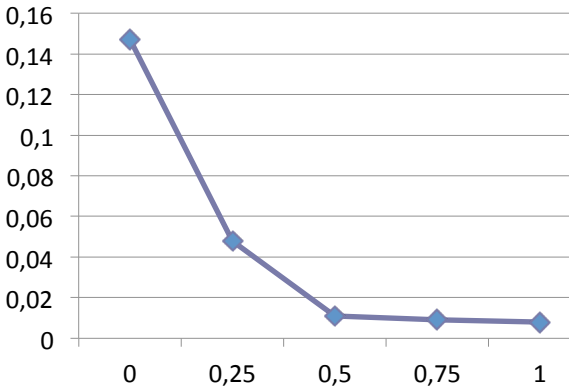


Рис. 4.44. Изменение вероятности обслуживать заявку более критического времени при изменении вероятности покинуть заявкой систему после стадии «Архитектура».

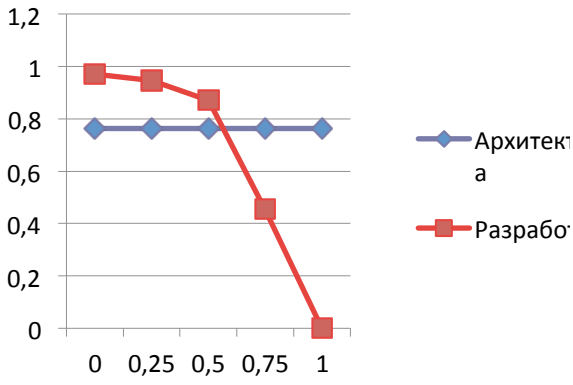


Рис. 4.43. Изменение средней загруженности всех исполнителей на обоих стадиях обслуживания при изменении вероятности покинуть заявкой систему после стадии «Архитектура».

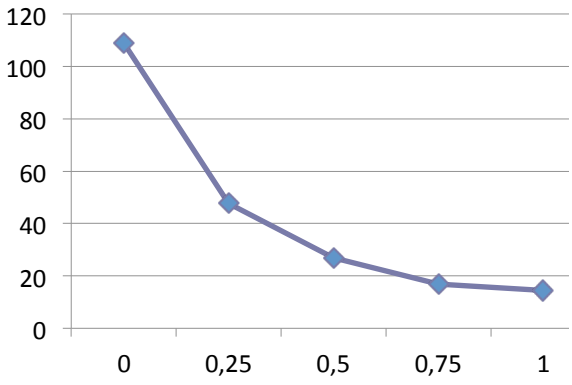


Рис. 4.45. Изменение оценки МО числа заявок в системе на конец недели при изменении вероятности покинуть заявкой систему после стадии «Архитектура».

Варьирование интенсивности обслуживания заявок исполнителями на обоих стадиях обслуживания.

В данном пункте всем исполнителям на всех стадиях обслуживания для любых типов заявок будет установлено одно значение интенсивности обслуживания. При этом закон распределения выбран экспоненциальным. При увеличении интенсивности обслуживания заявок исполнителями на обоих стадиях обслуживания критерии: оценка МО времени обслуживании заявок в системе, вероятность обслуживать заявку более критического времени, загрузка исполнителей и оценка МО числа заявок в системе на конец недели – должны уменьшиться. И наоборот, при уменьшении интенсивности обслуживания заявок исполнителями на обоих стадиях обслуживания все значения перечисленных критериев должны увеличиться (см. табл. 4.12 и рис. 4.46-4.49). Время моделирования составляет 180 суток. Критическое время составляет 30 суток.

Табл. 4.12.

Изменение критериев оценки моделирования в результате варьирования интенсивности обслуживания заявок исполнителями на обоих стадиях обслуживания.

Критерий	**Интенсивность обслуживания заявок ОУ на обоих стадиях обслуживания**			
	0,25	**0,5**	**2**	**4**
Оценка МО времени обслуживании заявок в системе	228,550	42,924	12,48	10,838
Вероятность обслуживаться заявку более критического времени	0,003	0	0	0
Средняя загруженность всех ОУ на стадии «Архитектура»	0,959	0,596	0,126	0,059
Средняя загруженность всех ОУ на стадии «Разработка»	0,392	0,251	0,074	0,037
Оценка МО числа заявок в системе на конец недели	50,846	8,385	2,538	2,231

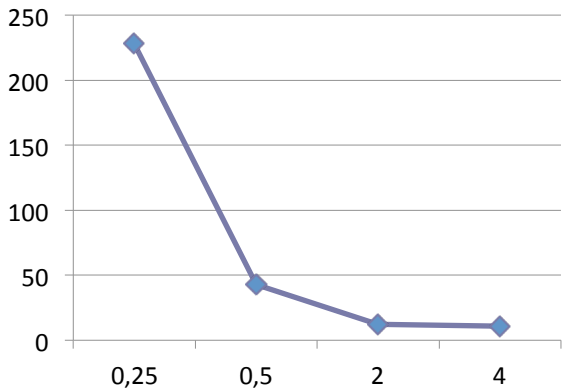


Рис. 4.46. Изменение оценки МО времени обслуживании заявок в системе при изменении интенсивности обслуживания заявок исполнителями на обоих стадиях обслуживания.

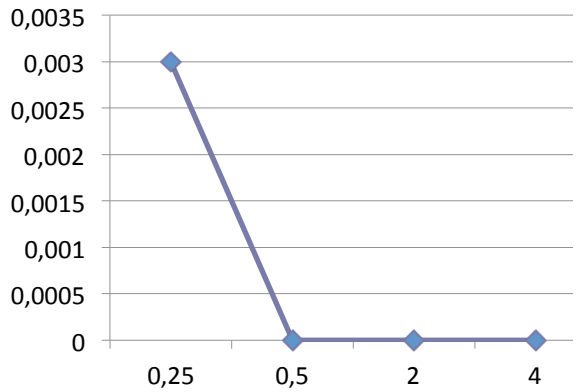


Рис. 4.48. Изменение вероятности обслуживать заявку более критического времени при изменении интенсивности обслуживания заявок исполнителями на обоих стадиях обслуживания.

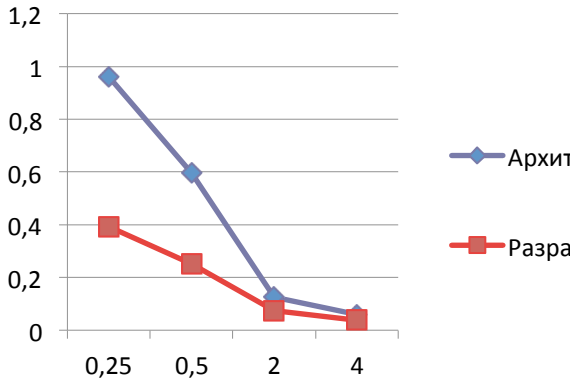


Рис. 4.48. Изменение средней загруженности всех исполнителей на обоих стадиях обслуживания при изменении интенсивности обслуживания заявок исполнителями на обоих стадиях обслуживания.

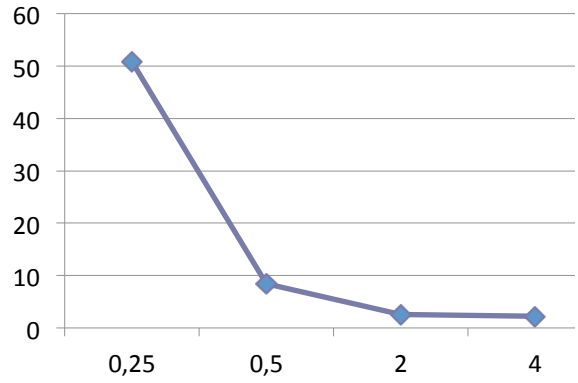


Рис. 4.49. Изменение оценки МО числа заявок в системе на конец недели при изменении интенсивности обслуживания заявок исполнителями на обоих стадиях обслуживания.

Варьирование интенсивности входного потока.

Возможность непосредственно задавать интенсивность входного потока в модели не предусмотрена. В модели делается различие между будними днями и выходными. Количество будних дней равняется 5, а выходных – 2. Интенсивность входного потока в будни существенно превосходит интенсивность входного потока в выходные дни. Следующие комбинации входных потоков соответствующие пяти и двухдневному интервалу позволяют получить четыре вида входных потоков, причем, чем номер больше, тем выше интенсивность входного потока:

1. 7 дней с интенсивностью входного потока как в выходной день.
2. 5 дней с интенсивностью входного потока как в выходной день и 2 дня с интенсивностью входного потока как в будний день.
3. 5 дней с интенсивностью входного потока как в будний день и 2 дня с интенсивностью входного потока как в выходной день.
4. 7 дней с интенсивностью входного потока как в будний день.

При увеличении интенсивности входного потока критерии: оценка МО времени обслуживании заявок в системе, вероятность обслуживать заявку более критического времени, загрузка исполнителей и оценка МО числа заявок в системе на конец недели – должны увеличиться. И, наоборот, при уменьшении интенсивности входного потока все значения перечисленных критериев должны уменьшиться (см. табл. 4.13 и рис. 4.50-4.53). Время моделирования составляет 180 суток. Критическое время составляет 30 суток.

Табл. 4.13.
Изменение критериев оценки моделирования в результате варьирования интенсивности входного потока.

Критерий	Интенсивность входного потока			
	1	2	3	4
Оценка МО времени обслуживании заявок в системе	39,713	106,878	128,073	372,445
Вероятность обслуживаться заявку более критического времени	0	0,018	0,024	0,229
Средняя загруженность всех исполнителей на стадии «Архитектура»	0,008	0,510	0,860	0,962
Средняя загруженность всех исполнителей на стадии «Разработка»	0,029	0,420	0,781	0,848
Оценка МО числа заявок в системе на конец недели	0,153	3,923	25,423	112,846

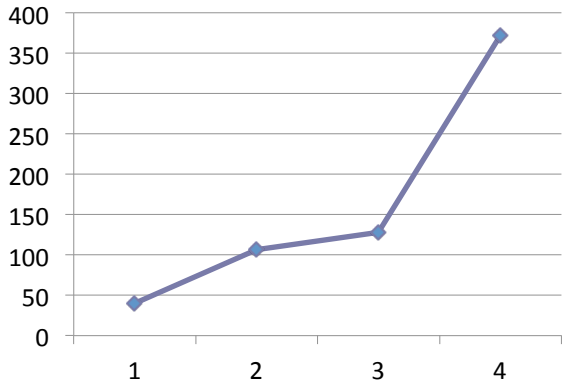


Рис. 4.50. Изменение оценки МО времени обслуживании заявок в системе при изменении интенсивности входного потока.

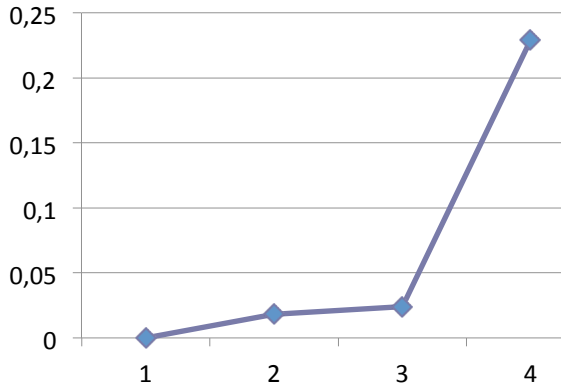


Рис. 4.52. Изменение вероятности обслуживать заявку более критического времени при изменении интенсивности входного потока.

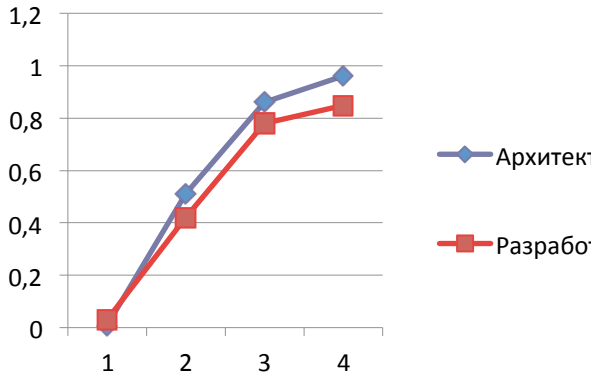


Рис. 4.52. Изменение средней загруженности всех исполнителей на обоих стадиях обслуживания при изменении интенсивности входного потока.

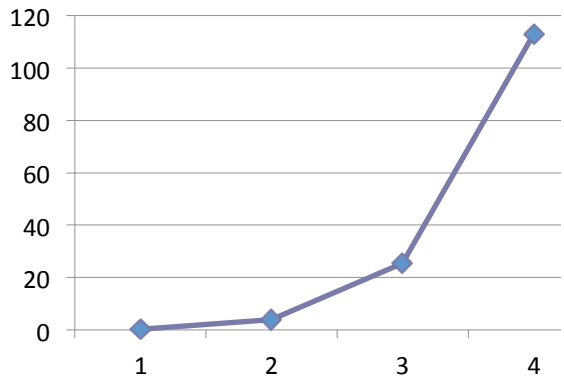


Рис. 4.53. Изменение оценки МО числа заявок в системе на конец недели при интенсивности входного потока.

Выводы по верификации модели.

При верификации произведен анализ изменения критериев оценки моделирования в ответ на изменения входных параметров. Изменению подверглись следующие параметры модели:

1. Количество исполнителей.
2. Вероятность покинуть заявкой систему после стадии «Архитектура».
3. Интенсивность обслуживания заявок исполнителями на обоих стадиях обслуживания.
4. Интенсивность входного потока.

Исследованию подверглись следующие критерии оценки моделирования:

5. Оценка МО времени обслуживании заявок в системе.
6. Вероятность обслуживаться заявку более критического времени.
7. Средняя загруженность всех исполнителей на стадии «Архитектура».
8. Средняя загруженность всех исполнителей на стадии «Разработка».
9. Оценка МО числа заявок в системе на конец недели.

При варьирование приведенных выше параметров изменение критериев оценок результатов моделирования совпало с ожидаемым. Это позволяет сделать вывод, что система правильно реагирует на изменения входных параметров.

4.11. Выводы по четвертой главе

1. Разработаны принципы проектирования СППР, которые могут использоваться широким кругом разработчиков. Эти принципы позволяют работать с векторным критерием высокой размерности и объединять различные методы теории принятия решений.
2. Разработана унифицированная концептуальная модель реляционной базы данных СППР, инвариантная по отношению к предметной области, позволяющая проводить настройку на конкретную задачу без изменения структуры базы данных.
3. Разработан каркасный подход к трехуровневой архитектуре программного обеспечения СППР, позволяющий проводить быструю и эффективную адаптацию оболочки СППР к решаемой задаче с учетом предпочтений ЛПР, а также проводить быстрое внедрение СППР с интеграцией в информационную среду предприятий.
4. Создан набор рекомендаций, обеспечивающих получение качественного программного обеспечения СППР при решении конкретных задач.
5. Разработаны инструментальные средства поддержки процесса управления разработкой СППР, в том числе имитационная модель для получения вероятностно-временных оценок производительности сотрудников занятых в разработке ПО СППР.

Глава 5. Создание СППР «Космос» на основе методологии унифицированной разработки СППР

Ниже рассмотрены вопросы информатизации и автоматизации формирования Этапной программы (ЭП) научно-прикладных исследований и экспериментов, планируемых на российском сегменте МКС. Осуществлена разработка СППР «Космос» для определения приоритетности космических экспериментов и разработана методика составления плана на основе предпочтений пользователя. СППР «Космос» использует все методологические рекомендации по разработке программного обеспечения СППР, изложенные в четвертой главе, а также унифицированную концептуальная модель реляционной БД СППР.

В настоящее время, в связи с увеличением количества предложений по новым космическим экспериментам (КЭ), а также КЭ, находящимся в стадии подготовки и реализации, актуальным является повышение эффективности экспертных процедур при выявлении приоритетов и формировании ЭП научно-прикладных исследований и экспериментов, планируемых на российском сегменте (РС) МКС. Достигнуть этого возможно путем применения информационных, в том числе сетевых, технологий на стадии принятия решений о включении проектов в ЭП [54, 57, 100-107, 112, 119, 123-128, 140].

На базе СППР DSS/UTES была создана новая СППР «Космос» позволяющая определять приоритетность КЭ. Работа с ней состоит из следующих этапов:

- ввод векторного критерия для оценки КЭ;
- ввод экспертами оценок конкретных КЭ по заданному векторному критерию;
- ввод системы ценностей, отражающей важности конкретных компонент векторного критерия (с помощью метода парных сравнений, зависимых предпочтений или иными формальными методами);
- вычисление обобщенных оценок КЭ на базе введенной информации в автоматическом режиме.

Для вычисления обобщённых оценок КЭ применяются как формальные методы на основе взвешенной суммы и парных сравнений, так и гибридный метод задания предпочтений, работа которого показана на рисунке 5.1.

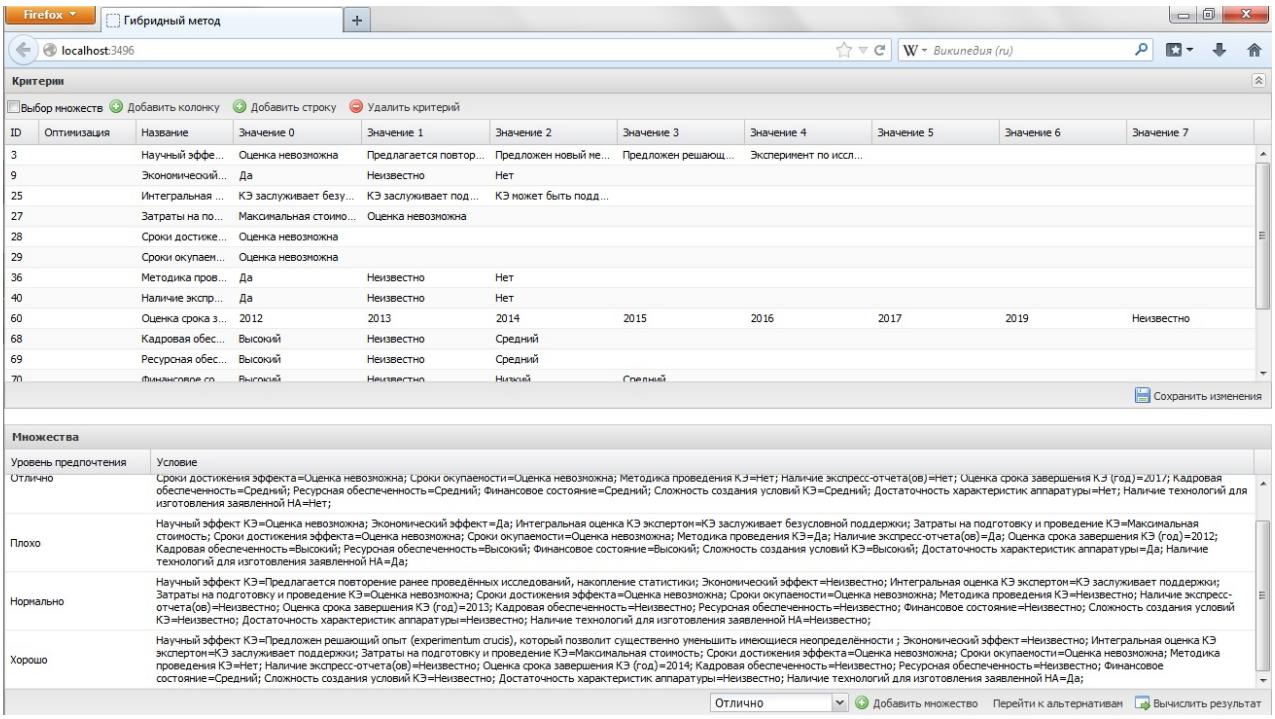


Рис. 5.1. Гибридный метод вычисления оценок КЭ.

Обобщенные оценки альтернатив позволяют перейти к следующему этапу работы с системой – это составление плана КЭ. Исходными данными для задачи служат ресурсные ограничения и ранжированный по значению обобщенной оценки список космических экспериментов, полученный в результате обработки экспертных оценок. Пример интерфейса СППР «Космос» с результатами ранжирования приведен на рисунке 5.2.

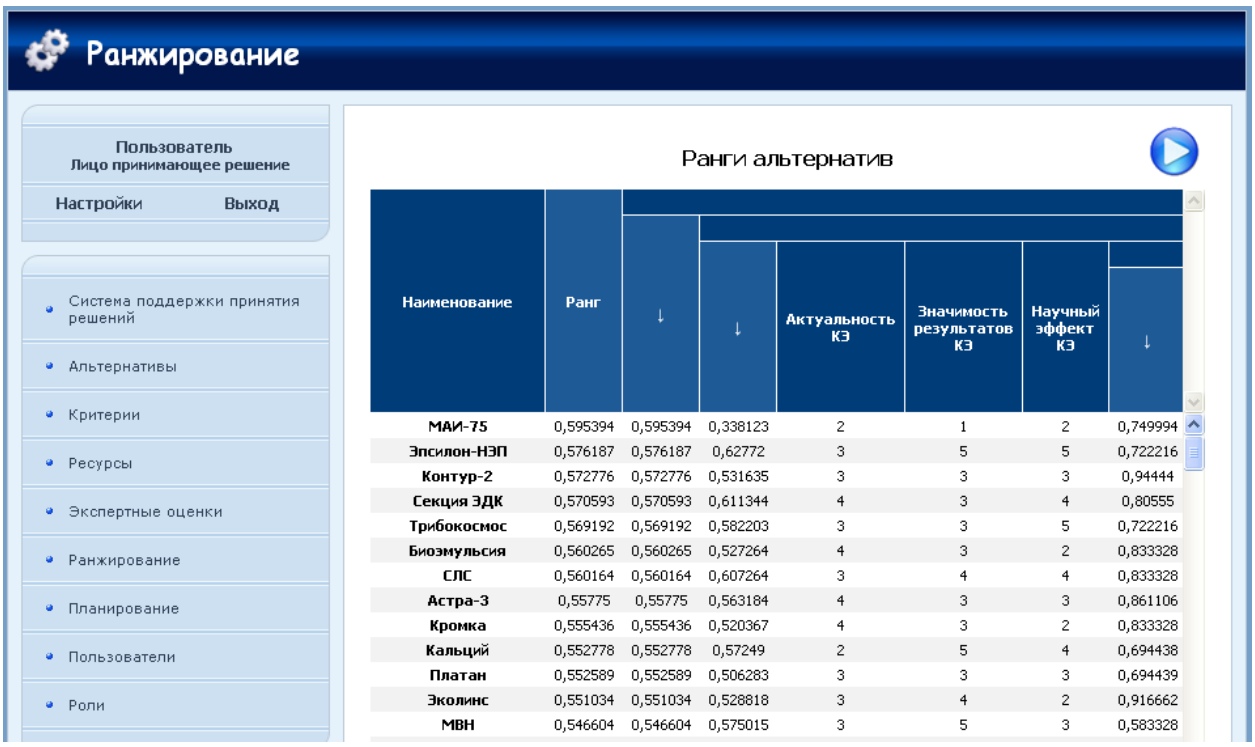


Рис. 5.2. Результаты ранжирования КЭ.

5.1. Постановка и решение задачи планирования

На заданном временном периоде необходимо спланировать последовательность проведения множества экспериментов на МКС, выбрав их из заданного множества возможных экспериментов и определив моменты их начала [100, 107]. Подходы к решению подобных задач изложены в работах Ю.И. Дегтярева [55-56].

Каждый КЭ характеризуется коэффициентом полезности (соответствует обобщенной оценке), длительностью, количеством необходимых для его проведения ресурсов различного вида как складируемого, так и не складируемого (мощностного) типа. Действует условие непрерывности проведения КЭ (начавшись, он не прерывается до его полного окончания). На проведение некоторых КЭ заданы условия выполнения из технологической последовательности (например, эксперимент *j* может начаться только после окончания экспериментов *k* и *l*).

Весь плановый период разбивается на дискретные составляющие (например, дни). На борту МКС имеется необходимый набор ресурсов обоих типов. Для мощностных ресурсов заданы их величины, не изменяемые по дням. Для складируемых ресурсов заданы величины, которые будут расходоваться на реализацию выбранных КЭ в течение некоторого интервала пополнения, заданного в выбранных временных единицах (например, в днях). Интенсивность расхода складируемых ресурсов предполагается равномерной в течение всего времени проведения эксперимента. После окончания очередного интервала пополнения неизрасходованные складируемые ресурсы восстанавливаются до заданного фиксированного значения. В таблице 5.1. приведены примеры ресурсов, с которыми работает система [132].

Табл. 5.1.

Ресурсы, с которыми работает система.

№	Название ресурса	Коли-чество	Период пополнения	Тип ресурса	Дискретность
1	Масса грузопотока вверх (кг.)	1000	365	складируемый	непрерывный
2	Масса грузопотока вниз (кг.)	100	365	складируемый	непрерывный
3	Объем грузопотока вверх ($м^3$)	1000	365	складируемый	непрерывный
4	Объем грузопотока вниз ($м^3$)	120	365	складируемый	непрерывный
5	Электроэнергия (кВт.ч.)	1200	365	складируемый	непрерывный
6	Пиковая мощность (кВт)	300		мощностной	дискретный
7	Посадочные места внутри	7		мощностной	дискретный
8	Посадочные места снаружи	5		мощностной	дискретный
9	Квота на время экипажа	8	1	складируемый	дискретный
10	Объём информации по каналам связи приём (Мбит)	10	1	складируемый	дискретный
11	Объём информации по каналам связи передача (Мбит)	3	1	складируемый	дискретный

Составленный план должен удовлетворять всем ресурсным ограничениям и быть оптимизирован по векторному критерию, включающему следующие составляющие:

1. Максимум суммарной обобщенной оценки включенных в план экспериментов.
2. Минимум средневзвешенного времени начала экспериментов.

Примем следующие обозначения:

J – общее количество возможных экспериментов для выбора;

j - номер эксперимента ($j \in J$);

q_{jk} - требуемый для проведения j-го эксперимента объем k-го складируемого ресурса (k= 1, 2,…, K);

V_l - величина l-го мощностного ресурса, имеющегося на МКС в каждый элементарный временной интервал;

τ_j - продолжительность j-го эксперимента (в выбранных единицах, например, в днях; предполагается, что все временные параметры и переменные определяются в этих единицах);

v_{jl} - требуемая для проведения j-го эксперимента величина l-го мощностного ресурса (l= 1, 2,…, L) на каждом элементарном временном интервале в течение времени его проведения;

T – продолжительность всего интервала планирования;

Δ - продолжительность интервала пополнения складируемых ресурсов;

t - текущий рассматриваемый элементарный временной интервал (1, 2,…, Δ, Δ+1, Δ +2,…, 2 Δ ,…, Δ (N-1)+1, Δ (N-1)+2,…, Δ N, где N – количество интервалов пополнения складируемых ресурсов на всем интервале планирования: T=Δ N);

Q_k - величина k-го складируемого ресурса, имеющегося на МКС в начале каждого интервала пополнения;

p_j - коэффициент полезности j-го эксперимента (оценивается на основе многомерной иерархической структуры факторов, учитываемых при оценке общей сравнительной полезности различных экспериментов). Далее предполагается, что номера экспериментов упорядочены по убыванию величин p_j с учетом ограничений на последовательность проведения экспериментов;

x_j - бинарная оптимизационная переменная, определяющая включение *j*-го эксперимента в формируемый план (=1, если планируется его включение, =0 в противном случае);

t_j - дискретная оптимизационная переменная, устанавливающая начало проведения *j*-го эксперимента;

$f_j(\tau) = \begin{cases} 1, & \textit{если } \tau \in [1, \tau_j] \\ 0 & \textit{в противном случае} \end{cases}$ - функция, определяющая факт проведения *j*-го эксперимента в произвольный рассматриваемый элементарный временной интервал t, определяя при этом $\tau = t - t_j + 1$ – как интервал времени с момента начала *j*-го эксперимента;

ps[j, s]=(*j, ps*[j, s] $\in J$) массив, определяющий перечень номеров экспериментов, предшествующих эксперименту с номером j.

На основе введенных обозначений определяются следующие выражения:

-затраты *k*-го складируемого ресурса для *j*-го эксперимента на интервале пополнения *n*:

$$\left(\sum_{t=(n-1)\Delta+1}^{n\Delta} f_j(t - t_j) \Big/ \tau_j \right) q_{jk};$$

- затраты *k*-го складируемого ресурса для всех экспериментов на интервале пополнения *n*:

$$\sum_{j=1}^{J} x_j \left\{ \left(\sum_{t=(n-1)\Delta+1}^{n\Delta} f_j(t - t_j) \Big/ \tau_j \right) q_{jk} \right\}$$

- величину используемого *j*-ым экспериментом *l*-го мощностного ресурса на элементарном временном интервале *t*:

$$f_j(t - t_j) v_{jl}$$

- величину используемого всеми включенными в план экспериментами *l*-го мощностного ресурса на элементарном временном интервале *t*:

$$\sum_{j=1}^{J} x_j f_j(t - t_j) v_{jl}$$

- ограничения на используемые складируемые ресурсы на всем интервале планирования:

$$\sum_{j=1}^{J} x_j \left\{ \left(\sum_{t=(n-1)\Delta+1}^{n\Delta} f_j(t-t_j) \Big/ \tau_j \right) q_{jk} \right\} \le Q_k, \qquad (k=1,2,\ldots,K;\ n=1,2,\ldots,N)$$

- ограничения на используемые мощностные ресурсы на всем интервале планирования:

$$\sum_{j=1}^{J} x_j f_j(t-t_j) v_{jl} \le V_l, (l=1,2,\ldots,L;\ t=1,2,\ldots,T)$$

- ограничения на технологическую последовательность проведения экспериментов, определяемую массивом *ps[j, s]*:

$$t_j \ge \max_s \{ s \in ps[j,s]: \quad t_p + \tau_p, \text{где } p = ps[j,s] \}$$

- ограничения, обеспечивающие синхронность изменения переменных x_j и t_j:

$$x_j \le t_j \le T x_j - \tau_j, \qquad (j=1,2,\ldots,J)$$

- суммарная полезность всех включенных в план экспериментов:

$$\sum_{j=1}^{J} x_j p_j$$

- суммарная взвешенная сумма моментов начала всех включенных в план экспериментов:

$$\sum_{j=1}^{J} x_j (t_j / p_j) \qquad (5.1)$$

- минимум выражения (5.1) соответствует максимуму следующего выражения:

$$\sum_{j=1}^{J} x_j p_j (T - t_j)/T$$

Таким образом, формализованная постановка задачи планирования будет иметь следующий вид:

$$\max_{x_j, t_j (j \in J)} \{ \sum_{j=1}^{J} x_j p_j;\ \sum_{j=1}^{J} x_j p_j (T - t_j)/T \}$$

$$\sum_{j=1}^{J} x_j \left\{ \left(\sum_{t=(n-1)\Delta+1}^{n\Delta} f_j(t-t_j) \Big/ \tau_j \right) q_{jk} \right\} \le Q_k, \qquad (k=1,2,\ldots,K;\ n=1,2,\ldots,N) \quad ,$$

$$\sum_{j=1}^{J} x_j f_j(t-t_j) v_{jl} \le V_l, (l=1,2,\ldots,L;\ t=1,2,\ldots,T),$$

$$t_j \ge \max_s \{ s \in ps[j,s]: \quad t_p + \tau_p, \text{где } p = ps[j,s] \},$$

$$x_j \le t_j \le T x_j - \tau_j, \qquad (j=1, 2,\ldots, J),$$

$x_j = 0$ *или* 1, $(j=1, 2,\ldots, J)$.

Данная задача планирования относятся к классу многокритериальных нелинейных задач дискретного целочисленного (комбинаторного по x_j) программирования. Наиболее подходящим методом решения данной задачи является метод неявного перебора, основанный на локальной стратегии поиска [37, 40, 55, 63, 73, 111, 118, 130, 138, 141, 161]. Этот метод, хотя и не гарантирует нахождения точного оптимального решения, как показывает опыт его использования, но позволяет находить с приемлемыми вычислительными затратами субоптимальные решения. Достоинством метода является возможность эффективного использования эвристик, учитывающих свойства решаемой задачи, при нахождении допустимых решений, при определении области перебора в районе рассматриваемой точки. Кроме того, существует возможность использования случайного механизма для реализации необходимой степени разнообразия при формировании начальных допустимых решений, что повышает вероятность выхода на оптимальное либо близкое к нему решение. На рисунке 5.3. показан фрагмент начального допустимого плана. В нем некоторые эксперименты не реализуются, т.к. ресурсов МКС не хватает для проведения всех экспериментов в заданном временном периоде.

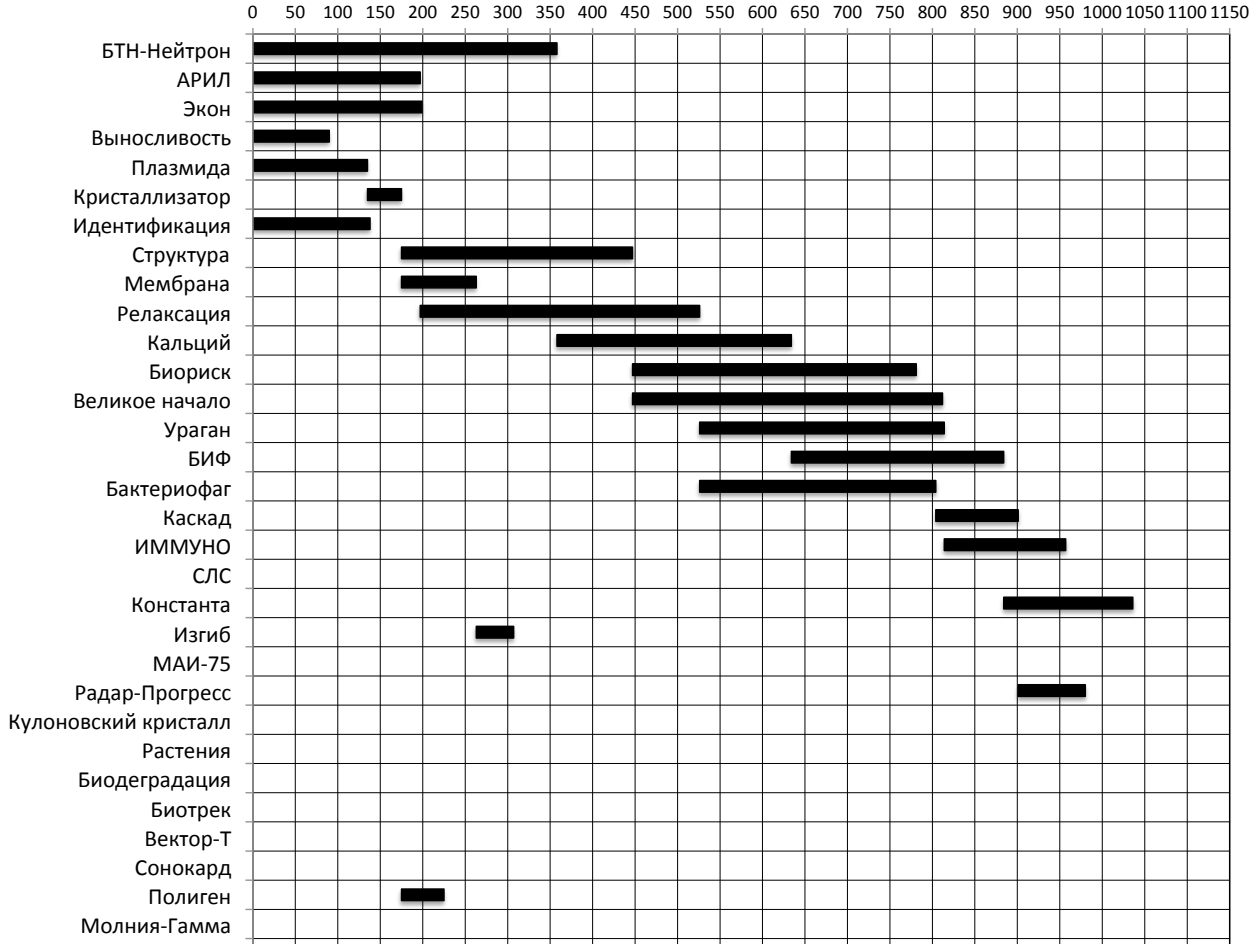


Рис. 5.3. Начальное допустимое решение.

Возможны следующие варианты стратегий улучшения начального допустимого решения:

1. Изъятие из плана некоторого эксперимента, освобождение занимаемых им ресурсов и включение в план эксперимента с большим значением p_j, которому теперь (после снятия эксперимента) ресурсов будет достаточно для включения в план.
2. Изъятие из плана некоторого эксперимента, освобождение занимаемых им ресурсов и включение в план двух или более экспериментов с меньшими значениями p_j (но суммарная полезность которых больше полезности исключаемого эксперимента), которым теперь (после снятия эксперимента) будет достаточно ресурсов для включения в план.
3. Изъятие из плана некоторого эксперимента, освобождение занимаемых им ресурсов и сдвижка влево (уменьшение) планируемых времен начала других экспериментов, включенных в план.

СППР «Космос» разработана в соответствии с методологией унифицированной разработки систем поддержки принятия многокритериальных решений. С СППР «Космос» должен работать широкий круг экспертов, который рассредоточен территориально по множеству городов и организаций [60-62, 68]. Поэтому СППР должна допускать сетевую модель работы в Intranet или Internet, так же необходимо минимизировать затраты на установку программного обеспечения на компьютеры пользователей. Для этого целесообразно использовать кроссплатфоменного тонкого клиента, т.е. клиента допускающего работу на компьютере с различными операционными системами. Кроме того, важно обеспечить соответствие HTML кода передаваемого на клиента последнему стандарту HTML5. Данным требованиям в полной мере удовлетворяет веб-браузер Mozilla Firefox, поэтому он и был выбран в качестве тонкого клиента. Веб-сервер работает под управлением IIS. Серверное приложение написано в среде Microsoft VisualStudio 2005 на C# с использованием ASP.NET. Программное обеспечение реализует предложенную унифицированную архитектуру, написано с использованием каркасного подхода и набора рекомендаций по обеспечению качества СППР. Взаимодействие с базой данных организовано исключительно через хранимые процедуры. Таким образом достигается оперативность изменений бизнес-логики, повышается производительность и гибкость, обеспечивается более высокий уровень безопасности. Физическая структура базы данных соответствует унифицированной концептуальной модели, предложенной в четвертой главе. В качестве СУБД использован MS SQL Server [136].

5.2. Реализация методологии унифицированной разработки в СППР «Космос»

5.2.1 Регистрация в системе

Для доступа к системе необходимо набрать в адресной строке браузера адрес, по которому находится СППР, – http://knts.tsniimash.ru/dss/. В результате появится главная страница СППР, рисунок 5.4.:

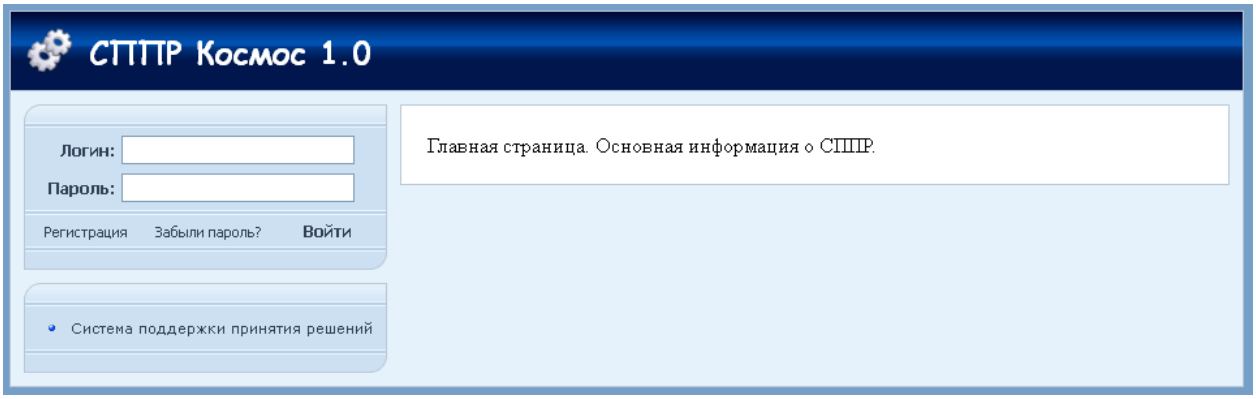


Рис. 5.4. Форма главной страницы СППР.

Для регистрации в системе надо нажать кнопку **Регистрация**. В результате отобразится форма регистрации пользователя, рисунок 5.5.:

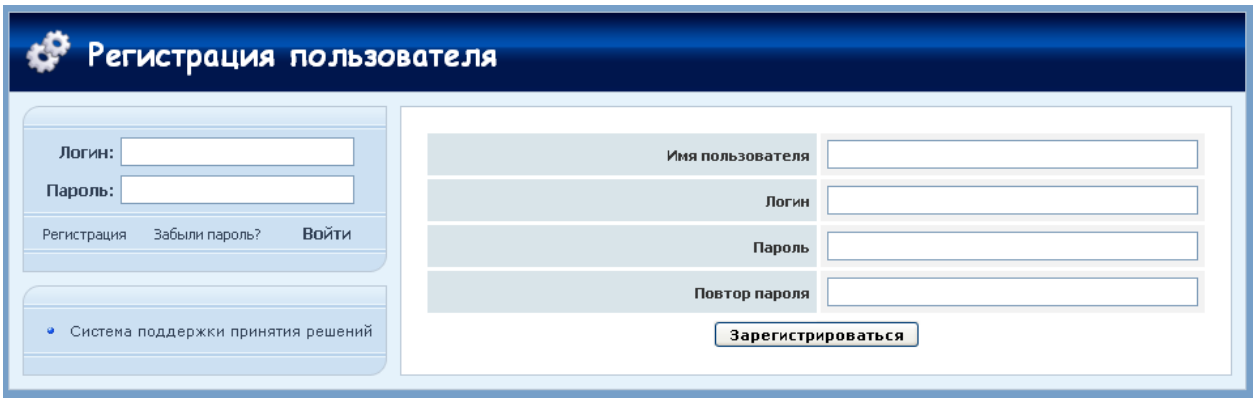


Рис. 5.5. Форма регистрации.

В данной форме необходимо ввести:

- имя пользователя – это ФИО пользователя;
- логин – краткое имя пользователя латинскими буквами;
- пароль – пароль не должен быть слишком простым и должен состоять из латинских букв и цифр;
- повтор пароля – необходимо в целях контроля правильности ввести пароль еще раз.

После регистрации в системе, требуется получить одобрение администратора системы и можно переходить к работе в СППР.

Для входа в систему необходимо ввести логин и пароль, указанные при регистрации, в левом верхнем углу основной формы СППР, рисунок 5.6.:

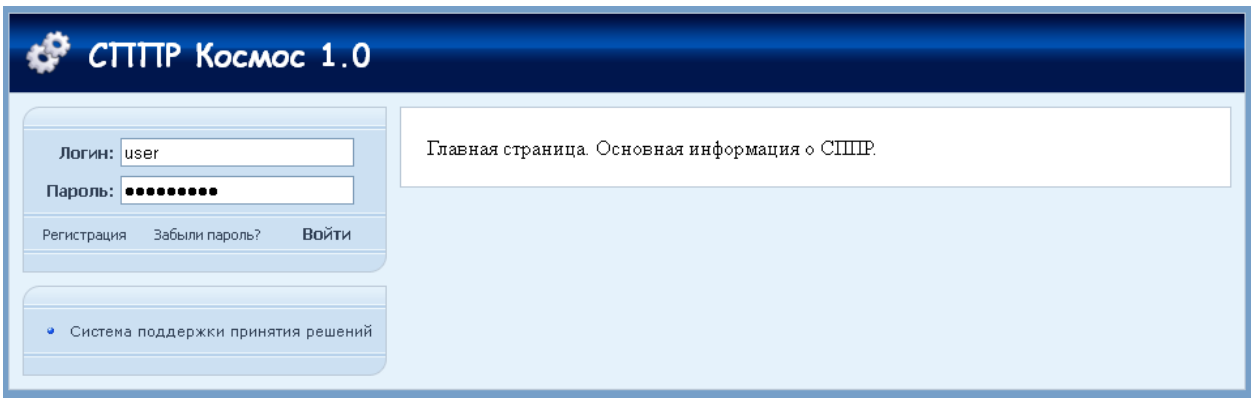


Рис. 5.6. Форма авторизации.

После ввода имени и пароля следует нажать кнопку **Войти**. Если пользователь забыл логин или пароль, то ему следует обратиться к администратору СППР.

5.2.2. Редактирование альтернатив

К задаче может быть привязано произвольное количество объектов оценки. Именно для них эксперты в дальнейшем вводят свои оценки. Эти объекты ранжируются, и среди них ЛПР выбирает рациональное решение. Эти объекты называются альтернативами.

Каждая альтернатива может быть привязана только к одной задаче. У одной задачи может быть сколько угодно альтернатив. Задачу, с которой работает пользователь, можно изменить в настройках системы, перейдя туда по нажатию на кнопку **Настройки** в левом верхнем меню.

Для редактирования альтернатив нужно нажать кнопку **Альтернативы** в левом меню. В результате будет показана страница редактора альтернатив, рисунок 5.7.:

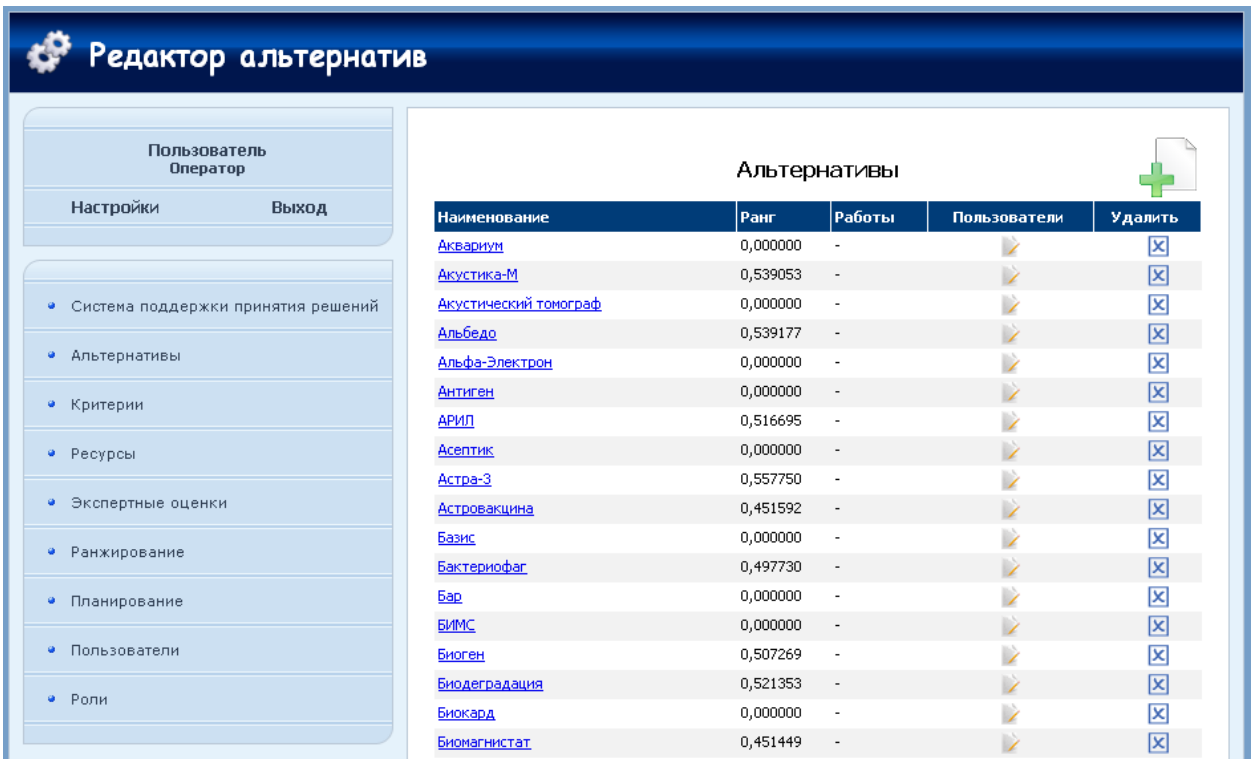


Рис. 5.7. Форма редактора альтернатив.

Для добавления новой альтернативы нужно нажать на кнопку в правом верхнем углу. В результате будет показана форма добавления альтернативы, рисунок 5.8.:

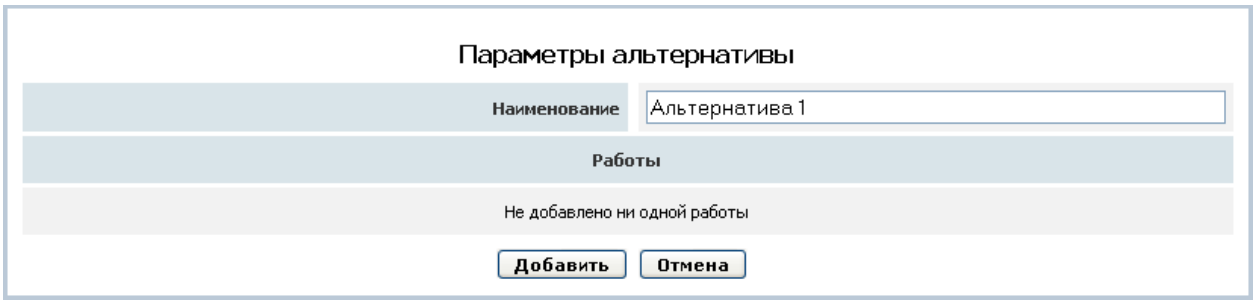


Рис. 5.8. Форма добавления альтернатив.

После ввода имени альтернативы нужно нажать кнопку **Добавить**. В результате форма добавления альтернативы исчезнет и новая альтернатива появится в списке альтернатив.

Для удаления альтернативы нужно нажать кнопку в соответствующей строке. Удалять альтернативы нужно с большой осторожностью, т.к. к альтернативе привязаны экспертные оценки и права доступа.

Ранги альтернатив вычисляются на основании экспертных оценок и предпочтений ЛПР и непосредственному редактированию не подлежат.

Для редактирования остальных параметров альтернативы, а также работ, присущих альтернативе, необходимо нажать на ссылку с именем редактируемой альтернативы из общего списка. При нажатии на ссылку откроется форма редактирования параметров альтернативы

(рисунок 5.9.), отличающаяся от формы добавления наличием кнопки в правой части формы, предназначенной для добавления работ.

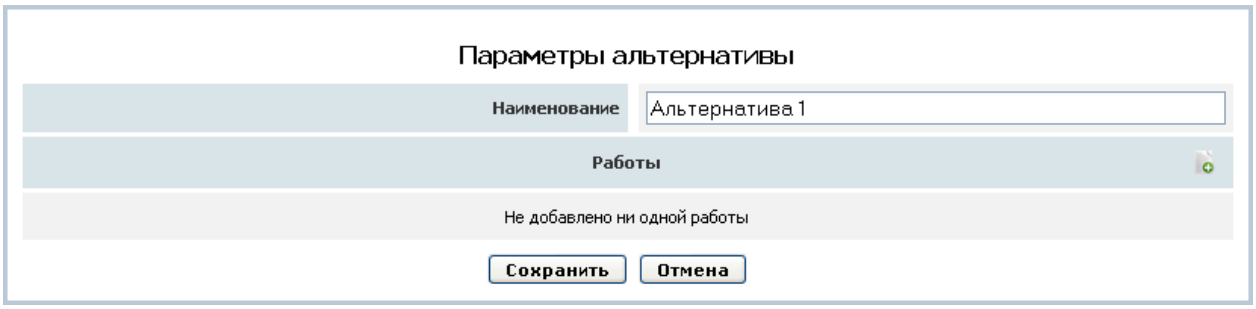


Рис. 5.9. Форма редактора альтернатив.

Для добавления работ текущей альтернативе необходимо нажать на кнопку в правой части формы. В результате будет показана форма добавления работы (КЭ), рисунок 5.10.:

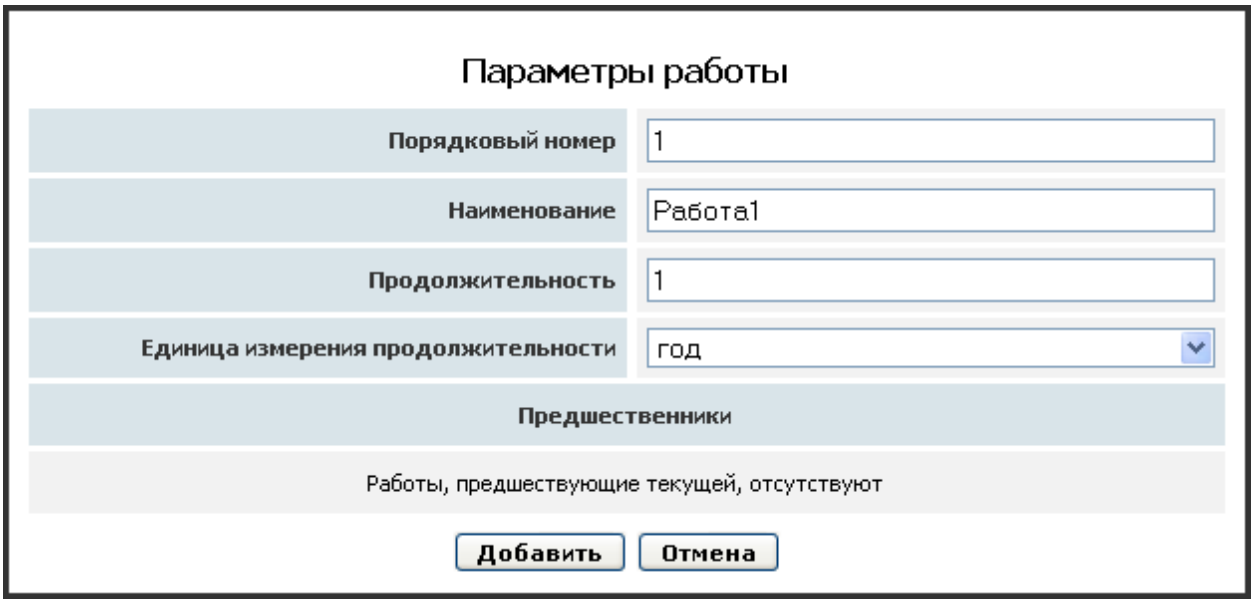


Рис. 5.10. Форма добавления работы (КЭ).

После заполнения всех полей необходимо нажать кнопку **Добавить**. В результате форма добавления работы исчезнет и новая работа появится в списке работ у текущей альтернативы, рисунок 5.11.:

Рис. 5.11. Форма текущих работ.

Для удаления работы нужно нажать кнопку в соответствующей строке.

Для редактирования параметров работы и ресурсов, которые пойдут на реализацию данной работы, необходимо нажать на ссылку с именем редактируемой работы из общего списка. При нажатии на ссылку откроется форма ввода параметров работы (рисунок 5.12.), отличающаяся от формы добавления наличием кнопки в правой части формы, предназначенной для добавления предшественников работ, а также наличием полей редактирования ресурсов:

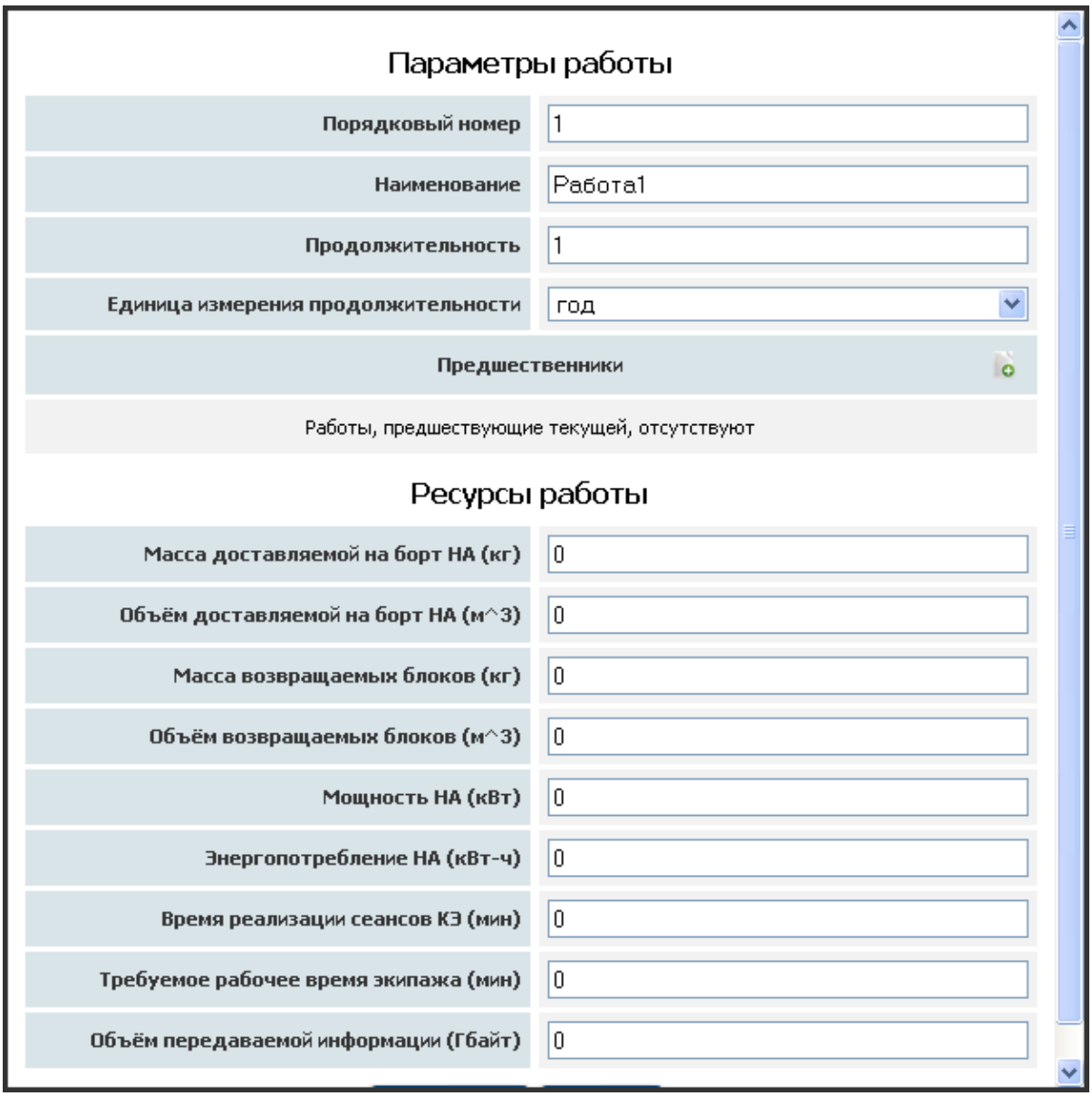


Рис. 5.12. Форма ввода параметров работы.

Для добавления предшественников работы текущей работе необходимо нажать на кнопку в правой части формы. В результате будет показана форма добавления предшественников работы, рисунок 5.13.:

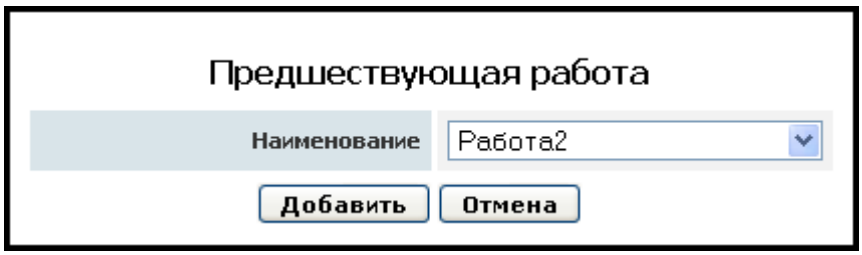


Рис. 5.13. Форма добавления предшествующей работы.

После выбора предшествующей работы в выпадающем списке необходимо нажать кнопку **Добавить**. В результате форма выбора предшествующей работы исчезнет и новая работа появится в списке работ предшественников у текущей работы, рисунок 5.14.:

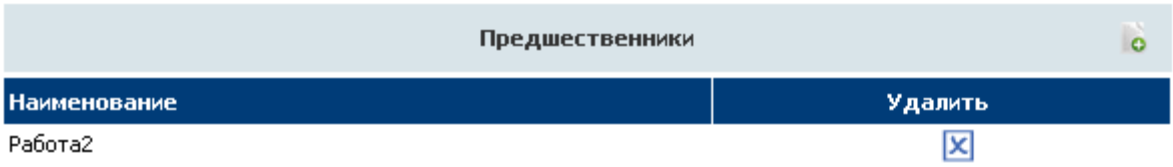


Рис. 5.14. Форма списка предшественников.

Для удаления предшествующей работы нужно нажать кнопку в соответствующей строке.

5.2.3. Редактирование критериев

Критерий – это атрибут альтернативы, участвующий в процедуре поддержки принятия решений.

Для редактирования критериев выберите на левой панели пункт меню **Критерии**. В результате отобразится страница редактора критериев, рисунок 5.15.:

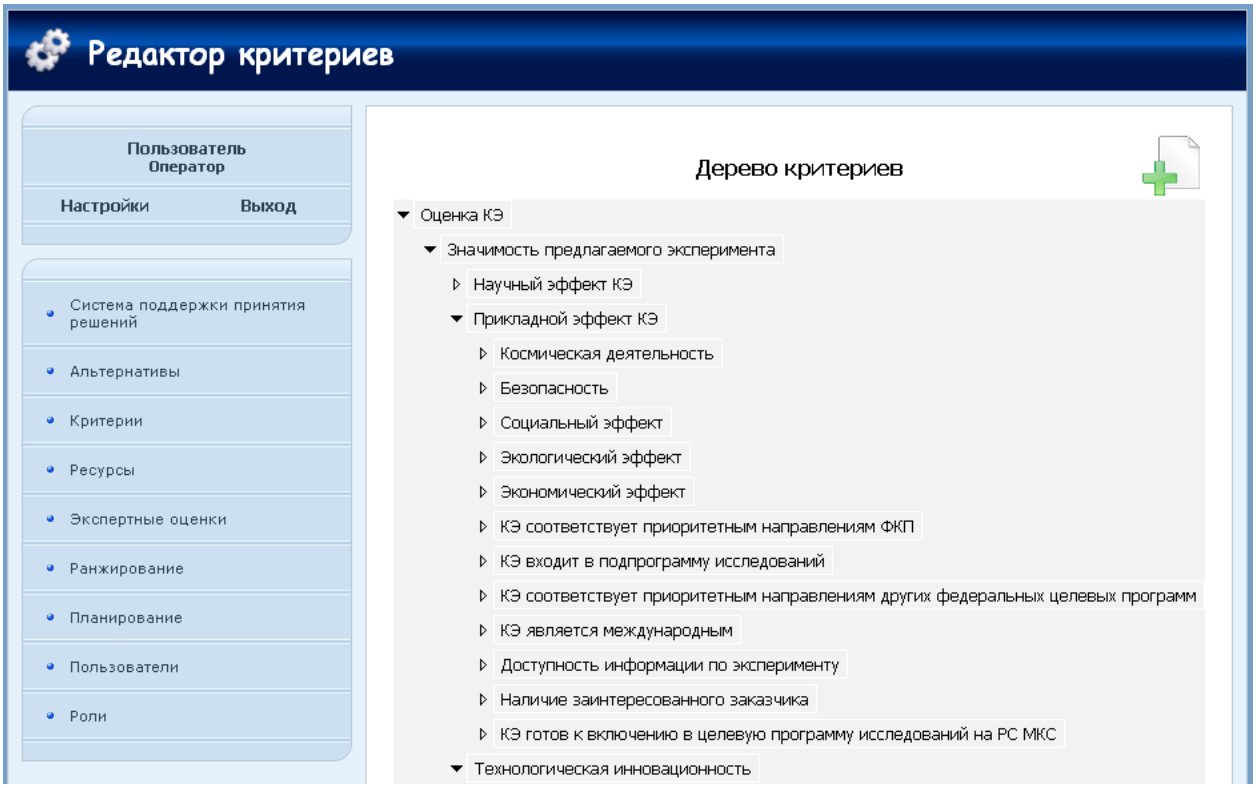


Рис. 5.15. Форма редактора критериев.

Дерево критериев может быть привязано только к одной задаче. У одной задачи может быть только одно дерево критериев. Задачу, с которой работает пользователь, можно изменить в настройках системы, перейдя туда по нажатию на кнопку **Настройки** в левом верхнем меню.

Для добавления нового критерия нужно нажать на кнопку в правом верхнем углу. В результате будет показана форма добавления критерия, рисунок 5.16.:

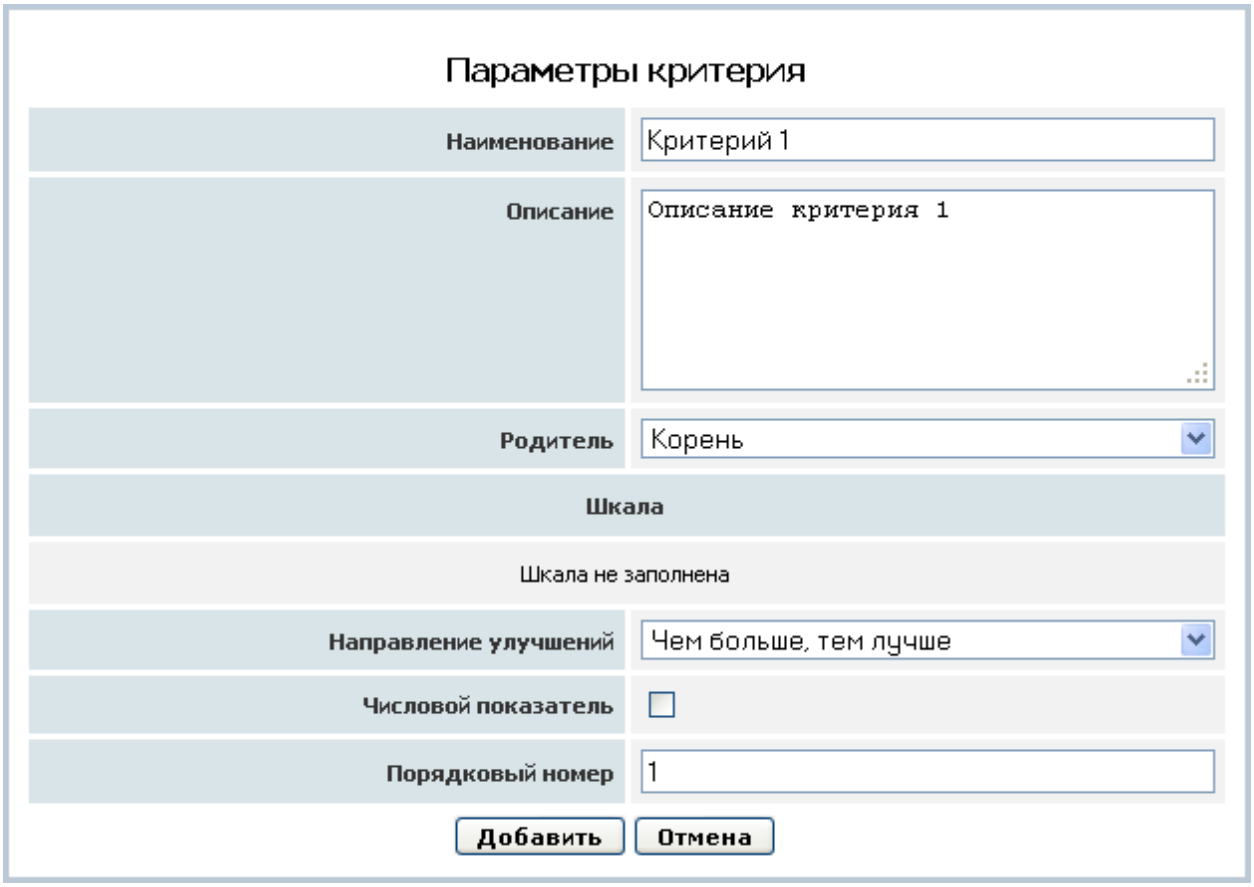


Рис. 5.16. Форма добавления критерия.

После ввода параметров критерия нужно нажать кнопку **Добавить**. В результате форма добавления критерия исчезнет и новый критерий появится в дереве.

Для редактирования параметров критерия, а также оценочной шкалы, присущей критерию, необходимо нажать на него в дереве. В результате откроется форма редактирования параметров критерия, отличающаяся от формы добавления наличием кнопки добавления пунктов оценочной шкалы в правой части формы и иконкой удаления критерия.

Для удаления критерия нужно нажать кнопку в правом нижнем углу на форме параметров критерия, представленной на рисунке 5.17.:

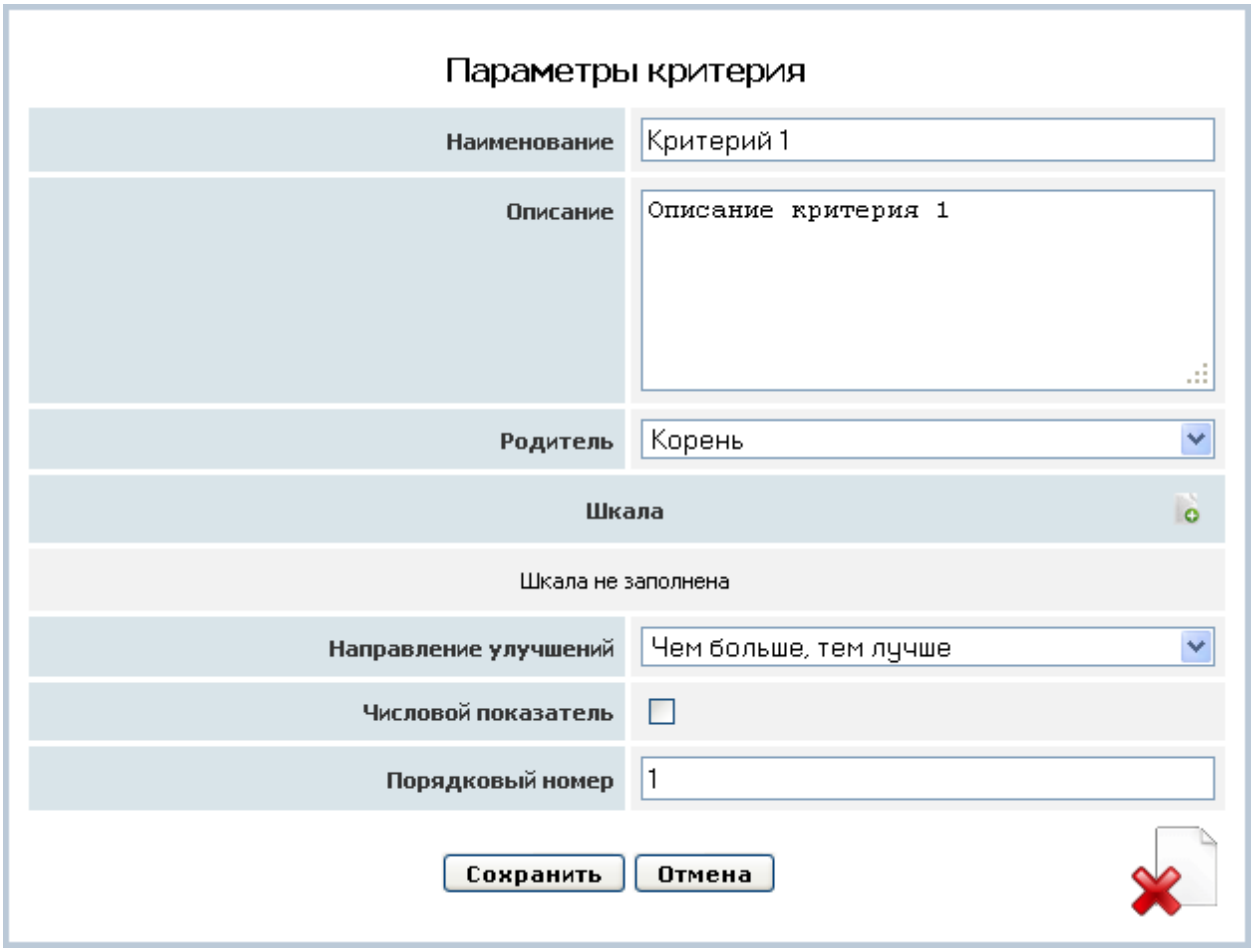


Рис. 5.17. Форма редактирования критерия.

Для добавления пунктов оценочной шкалы необходимо нажать на кнопку в правой части формы. В результате будет показана форма добавления и редактирования пунктов оценочной шкалы, рисунок 5.18.:

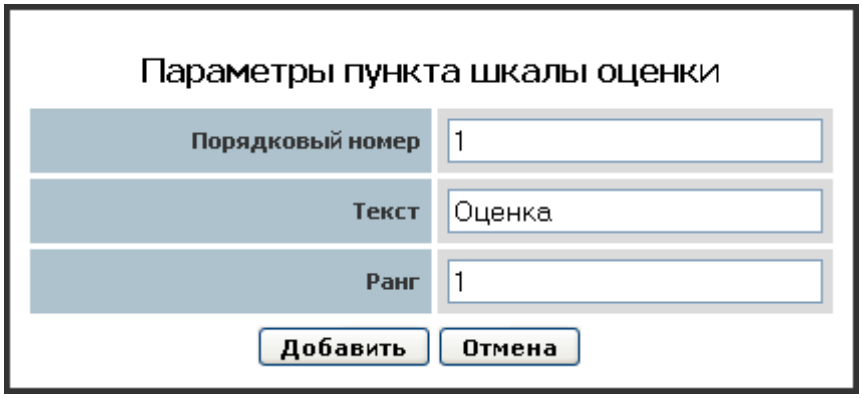


Рис. 5.18. Форма добавления и редактирования оценочной шкалы.

После заполнения всех полей необходимо нажать кнопку **Добавить**. В результате форма добавления пунктов оценочной шкалы исчезнет и новый пункт оценочной шкалы появится в списке у текущего критерия, рисунок 5.19.:

Рис. 5.19. Форма списка оценочных шкал критериев.

Для удаления пункта оценочной шкалы нужно нажать кнопку в соответствующей строке.

Допускается, что шкала оценки критерия может быть не заполнена, в этом случае критерий автоматически становится числовым.

Кроме редактирования шкалы оценки среди атрибутов критерия следует указать его остальные атрибуты:

- наименование;
- описание;
- родитель – критерий, стоящий на 1 ступень выше в иерархии дерева.

Направление улучшений, форма представлена на рисунке 5.20.:

- чем, больше, тем лучше (например, чем больше прибыль, тем лучше);
- чем меньше, тем лучше (например, чем меньше затраты, тем лучше);
- идеальное значение (например, комфортная температура в комнате 23^0C). Это идеальное значение можно ввести, если установить переключатель: «Порядковый номер» (необходим лишь для удобства сортировки).

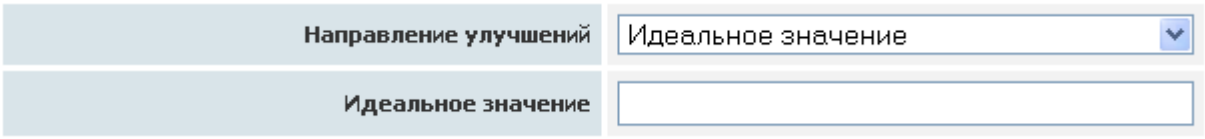


Рис. 5.20. Форма редактирования дополнительных атрибутов критериев.

После того как данные изменены у пользователя есть возможность сохранить изменения в текущем выбранном узле. Для этого нужно нажать на кнопку **Сохранить**, чтобы изменения вступили в силу.

В результате форма редактирования критерия исчезнет и новые параметры будут записаны в базу данных.

Если критерий обобщенный, т.е. у него есть потомки в дереве, то ЛПР следует задать метод свертки. Их должен задавать и редактировать ЛПР. Поэтому методы свертки описаны в инструкции ЛПР.

5.2.4. Редактирование ресурсов

Ресурс – это количественная мера, определяющая возможности включения в план тех или иных альтернатив.

Каждый ресурс может быть привязан только к одной задаче. У одной задачи может быть сколько угодно ресурсов. Задачу, с которой работает пользователь, можно изменить в настройках системы, перейдя туда по нажатию на кнопку **Настройки** в левом верхнем меню.

Для редактирования ресурсов нужно нажать кнопку **Ресурсы** в левом меню. В результате будет показана страница редактора ресурсов, рисунок 5.21.:

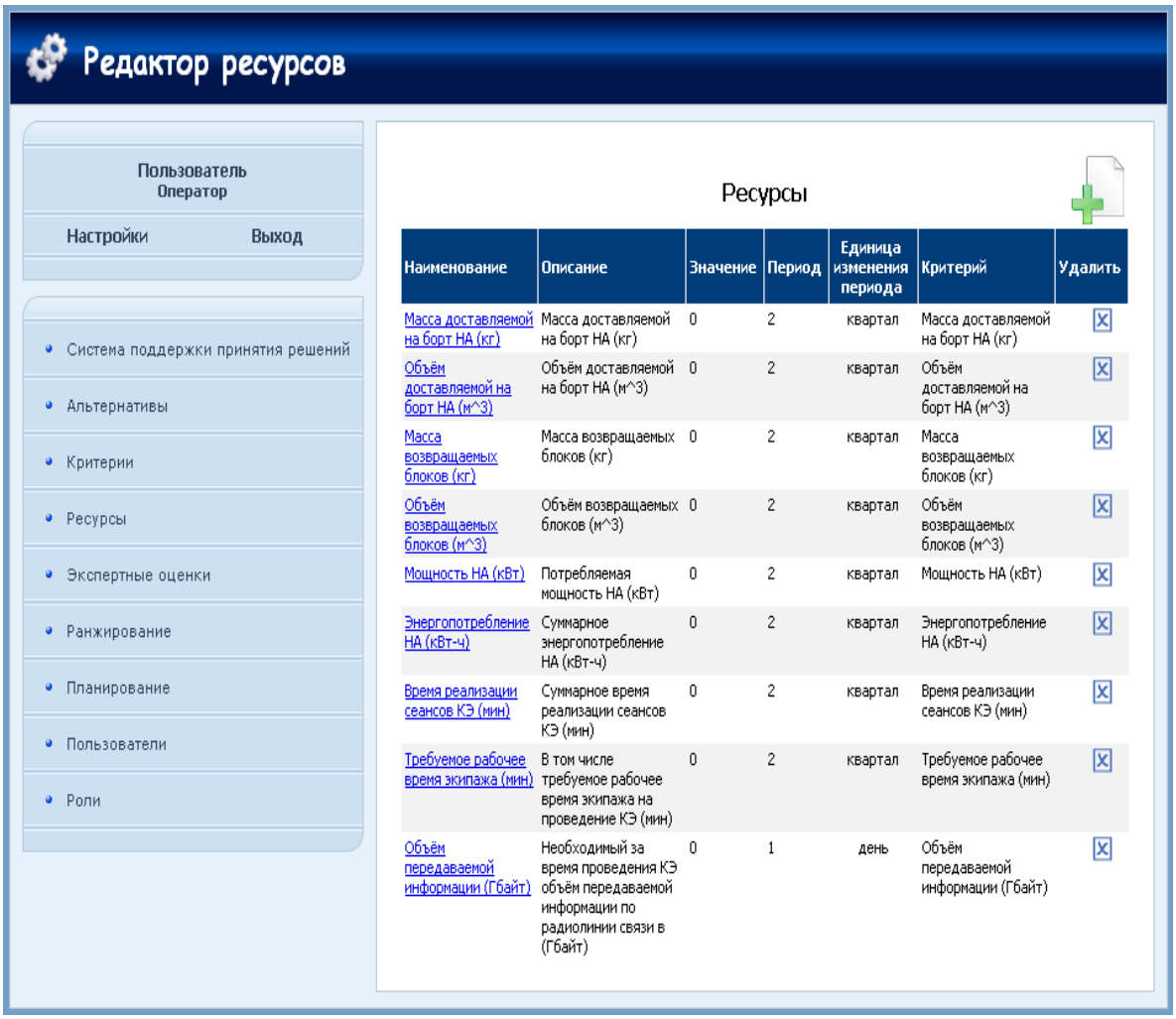


Рис. 5.21. Форма редактора ресурсов.

Для добавления нового ресурса нужно нажать на кнопку в правом верхнем углу. В результате будет показана форма добавления ресурса, рисунок 5.22.:

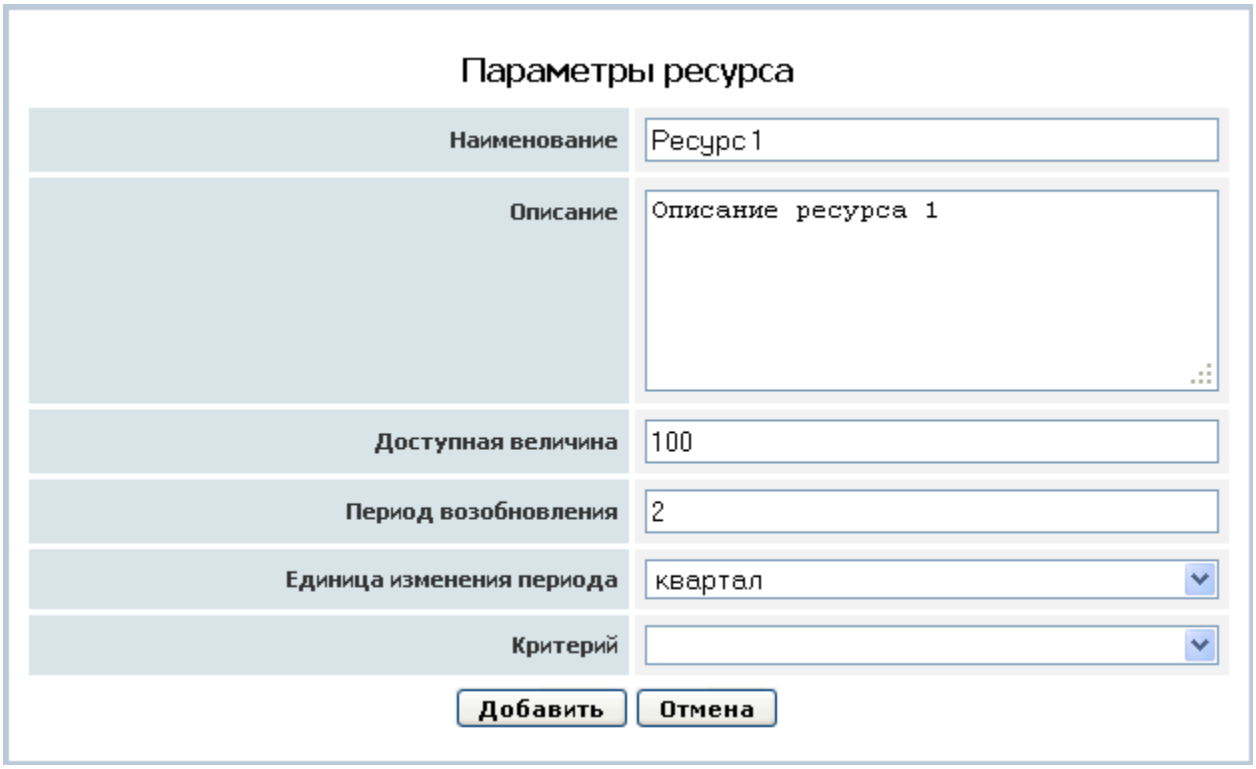


Рис. 5.22. Форма добавления ресурса.

После ввода параметров нужно нажать кнопку **Добавить**. В результате форма добавления ресурса исчезнет и новый ресурс появится в списке ресурсов.

Для удаления ресурса нужно нажать кнопку в соответствующей строке.

Для редактирования параметров ресурса необходимо нажать на ссылку с именем ресурса из общего списка. При нажатии на ссылку откроется форма редактирования параметров ресурса аналогичная форме добавления ресурса. При изменении параметров ресурса, необходимо нажать кнопку **Сохранить**, чтобы изменения вступили в силу. В результате форма редактирования параметров ресурса исчезнет и новые параметры ресурса будут записаны в базу данных.

5.2.5. Выбор методов

Для выбора метода свертки по агрегирующему критерию пользователю следует перейти на страницу критериев и выбрать требуемый узел в дереве, имеющий под собой потомков. В результате будет показана форма редактирования параметров критерия. Однако вместо параметра **Направление улучшений**, присущему не агрегирующему критерию, в форме будет присутствовать параметр **Метод агрегирования**, присущий агрегирующему критерию (рисунок 5.23.):

Рис. 5.23. Поле выбора метода агрегирования.

Установка требуемого метода агрегирования осуществляется путем соответствующего выбора в выпадающем списке.

5.2.6. Редактирование системы ценностей

Для каждого из методов свертки векторного критерия существует набор параметров описывающих систему ценностей пользователя. Для настройки этих параметров следует нажать на кнопку справа от выпадающего списка в графе **Метод агрегирования**. В результате откроется окно настройки соответствующего метода.

Для взвешенной суммы путем непосредственных назначений откроется страница, которая представлена на рисунке 5.24.:

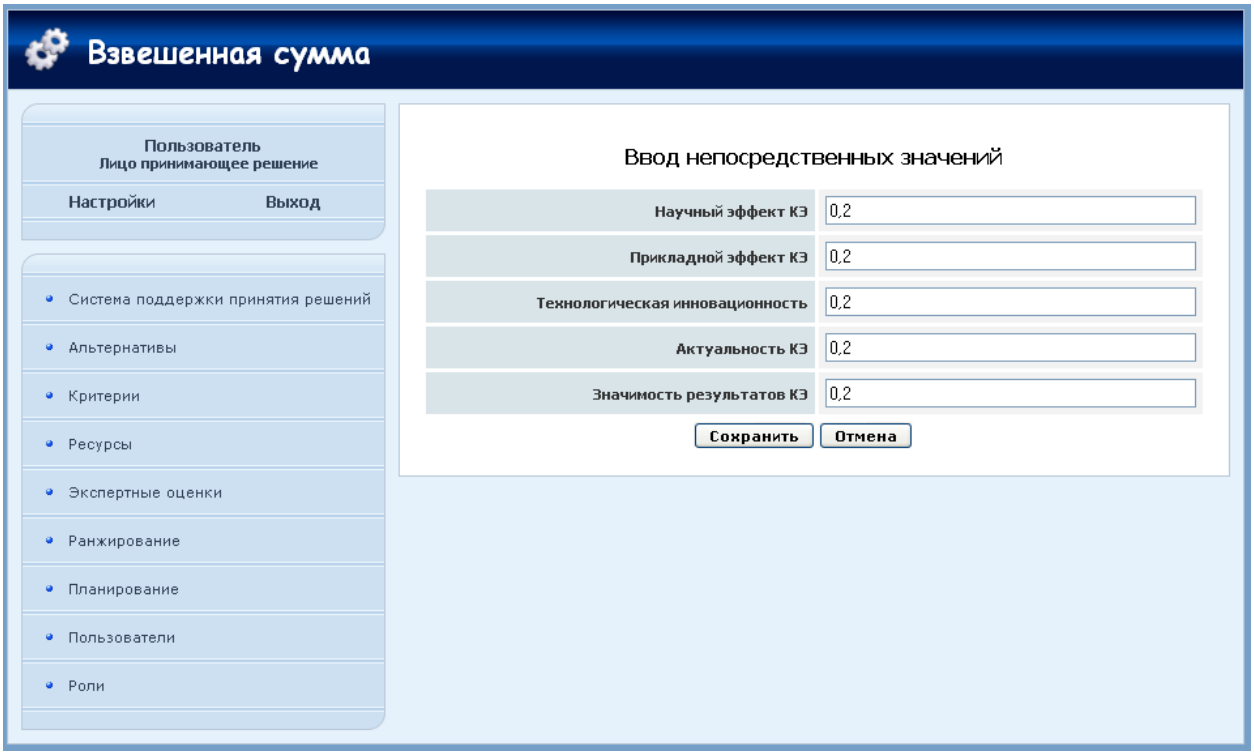


Рис. 5.24. Форма ввода предпочтений ЛПР.

В ней следует ввести или изменить веса критериев в соответствии с предпочтениями ЛПР. Чем больше важность критерия, тем выше его вес. В дальнейшем, при запуске процедуры ранжирования веса критериев будут нормированы так, что их сумма будет равна единице. Для сохранения результатов следует нажать кнопку **Сохранить**.

Если выбран метод парных сравнений, то откроется страница, рисунок 5.25.:

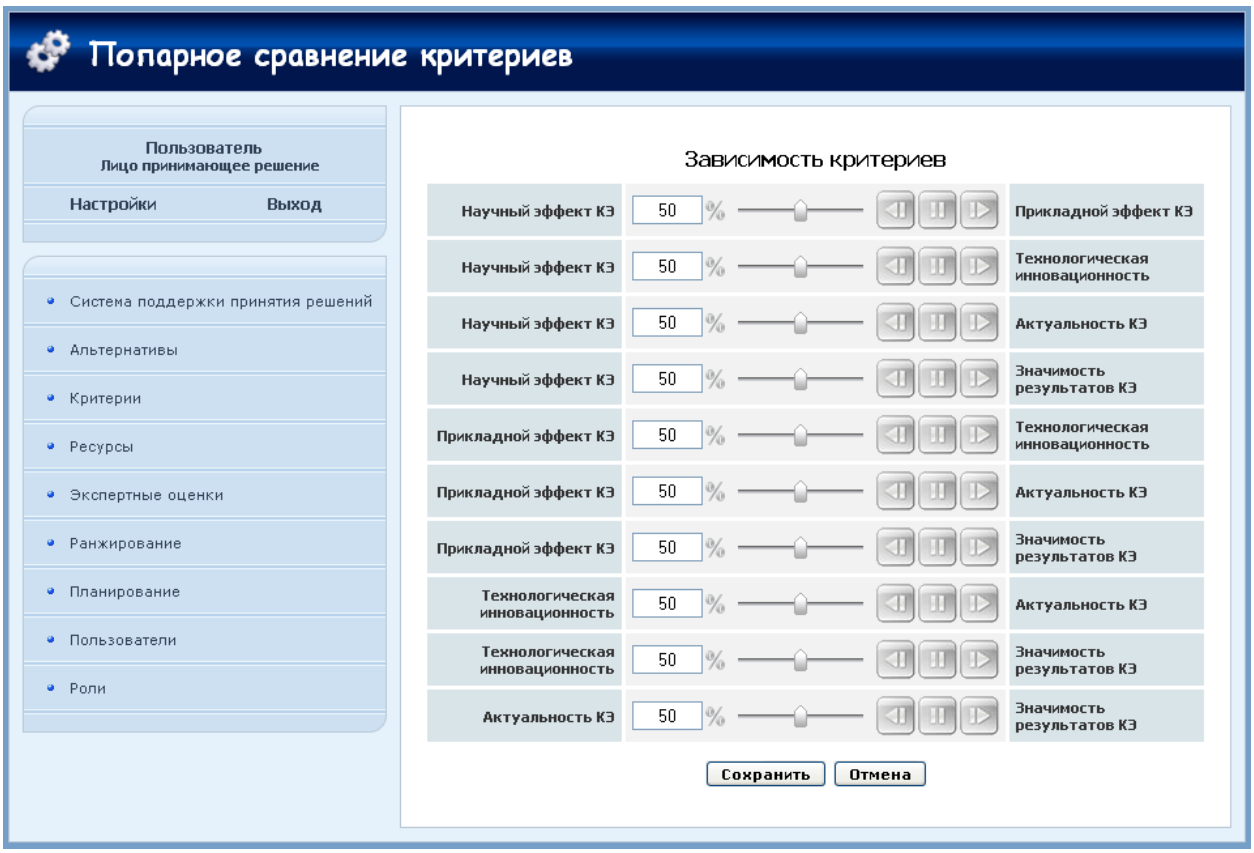


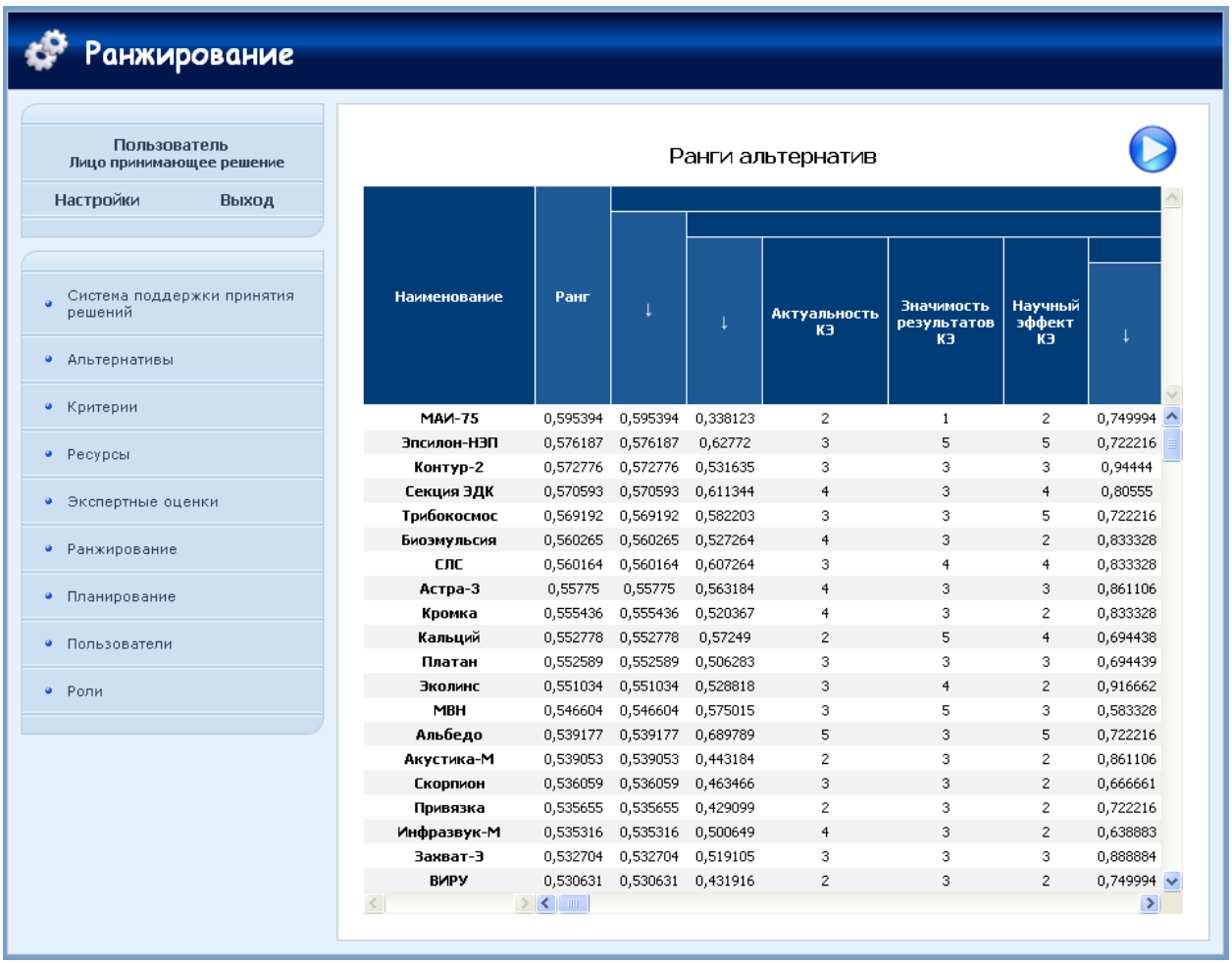
Рис. 5.25. Форма настройки метода парных сравнений критериев.

На данной странице требуется указать: насколько один критерий важнее другого. И так для всех комбинаций двух критериев. Для изменения соотношения следует подвинуть ползунок в сторону более важного критерия. После завершения редактирования следует нажать на кнопку **Сохранить**.

5.2.7. Запуск процедуры ранжирования и мониторинг результатов

Для запуска процедуры ранжирования следует выбрать в меню пункт **Ранжирование**. Ранжирование альтернатив осуществляется всегда в рамках одной задачи. Задачу, с которой работает пользователь, можно изменить в настройках системы, перейдя туда по нажатию на кнопку **Настройки** в левом верхнем меню. Перед ранжированием эксперты должны проставить каждой альтернативе не менее одной оценки по каждому из первичных критериев. Как проставляются эти оценки сказано в инструкции для экспертов.

Ниже показана страница Ранжирования альтернатив, рисунок 5.26.:



Наименование	Ранг	↓	↓	Актуальность КЭ	Значимость результатов КЭ	Научный эффект КЭ	↓
МАИ-75	0,595394	0,595394	0,338123	2	1	2	0,749994
Эпсилон-НЭП	0,576187	0,576187	0,62772	3	5	5	0,722216
Контур-2	0,572776	0,572776	0,531635	3	3	3	0,94444
Секция ЭДК	0,570593	0,570593	0,611344	4	3	4	0,80555
Трибокосмос	0,569192	0,569192	0,582203	3	3	5	0,722216
Биоэмульсия	0,560265	0,560265	0,527264	4	3	2	0,833328
СЛС	0,560164	0,560164	0,607264	3	4	4	0,833328
Астра-3	0,55775	0,55775	0,563184	4	3	3	0,861106
Кромка	0,555436	0,555436	0,520367	4	3	2	0,833328
Кальций	0,552778	0,552778	0,57249	2	5	4	0,694438
Платан	0,552589	0,552589	0,506283	3	3	3	0,694439
Эколинс	0,551034	0,551034	0,528818	3	4	2	0,916662
МВН	0,546604	0,546604	0,575015	3	5	3	0,583328
Альбедо	0,539177	0,539177	0,689789	5	3	5	0,722216
Акустика-М	0,539053	0,539053	0,443184	2	3	2	0,861106
Скорпион	0,536059	0,536059	0,463466	3	3	2	0,666661
Привязка	0,535655	0,535655	0,429099	2	3	2	0,722216
Инфразвук-М	0,535316	0,535316	0,500649	4	3	2	0,638883
Захват-Э	0,532704	0,532704	0,519105	3	3	3	0,888884
ВИРУ	0,530631	0,530631	0,431916	2	3	2	0,749994

Рис. 5.26. Форма ранжирования альтернатив.

Для запуска процедуры ранжирования следует нажать кнопку . В результате в колонке Ранг появятся посчитанные значения. Столбцы озаглавленные стрелкой, показывающей вертикально вниз, показывают рассчитанное значение агрегирующего критерия, охватывающего группу остальных критериев (своих потомков) справа от стрелки. У этой таблицы есть одно важное свойство: она динамична. Рассмотрим ситуацию, когда таблица настолько разрастается от количества альтернатив или от количеств секций, что не помещается на экран целиком. Тогда, для того, чтобы видеть шапку таблицы, название альтернатив и её оценок, необходимо схватиться мышкой за поле с оценками и передвинуть его туда, куда нам будет угодно.

5.2.8. **Редактирование** списка планов

План – это распределение подмножества альтернатив на временной шкале.

Для редактирования планов на левой панели имеется пункт меню **Планирование**. В результате отобразится страница планирования программ, рисунок 5.27.:

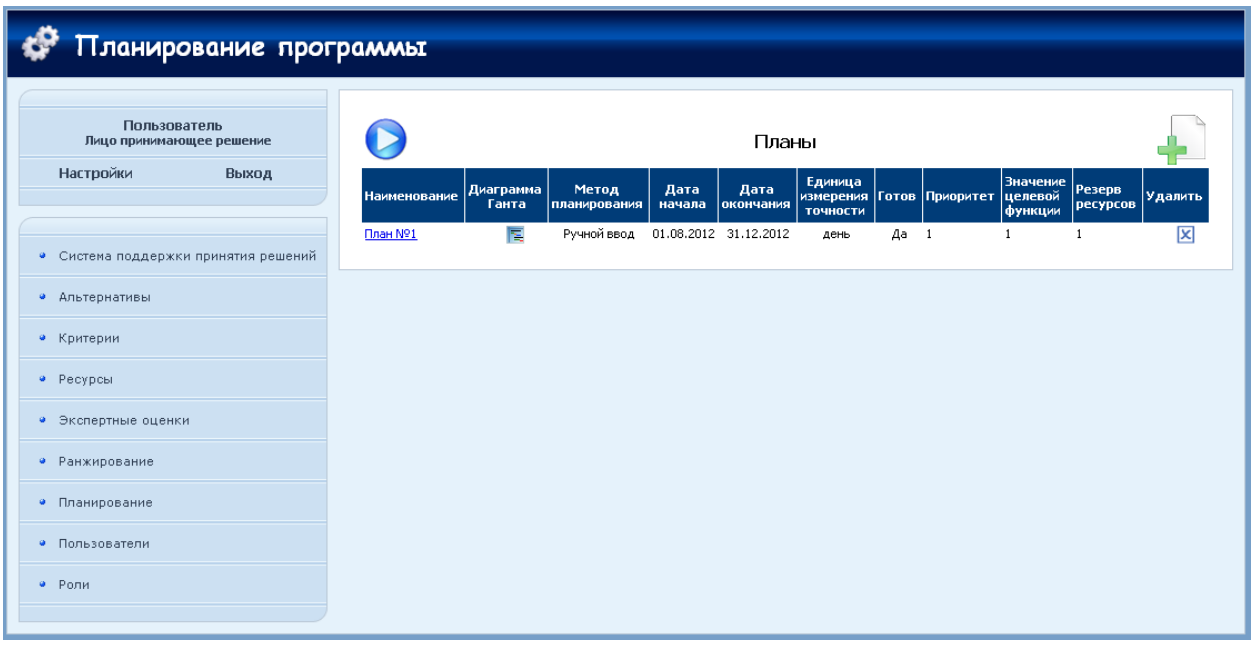


Рис. 5.27. Форма планирования программы.

Для добавления нового плана нужно нажать на кнопку в правом верхнем углу. В результате будет показана форма добавления плана, рисунок 5.28.:

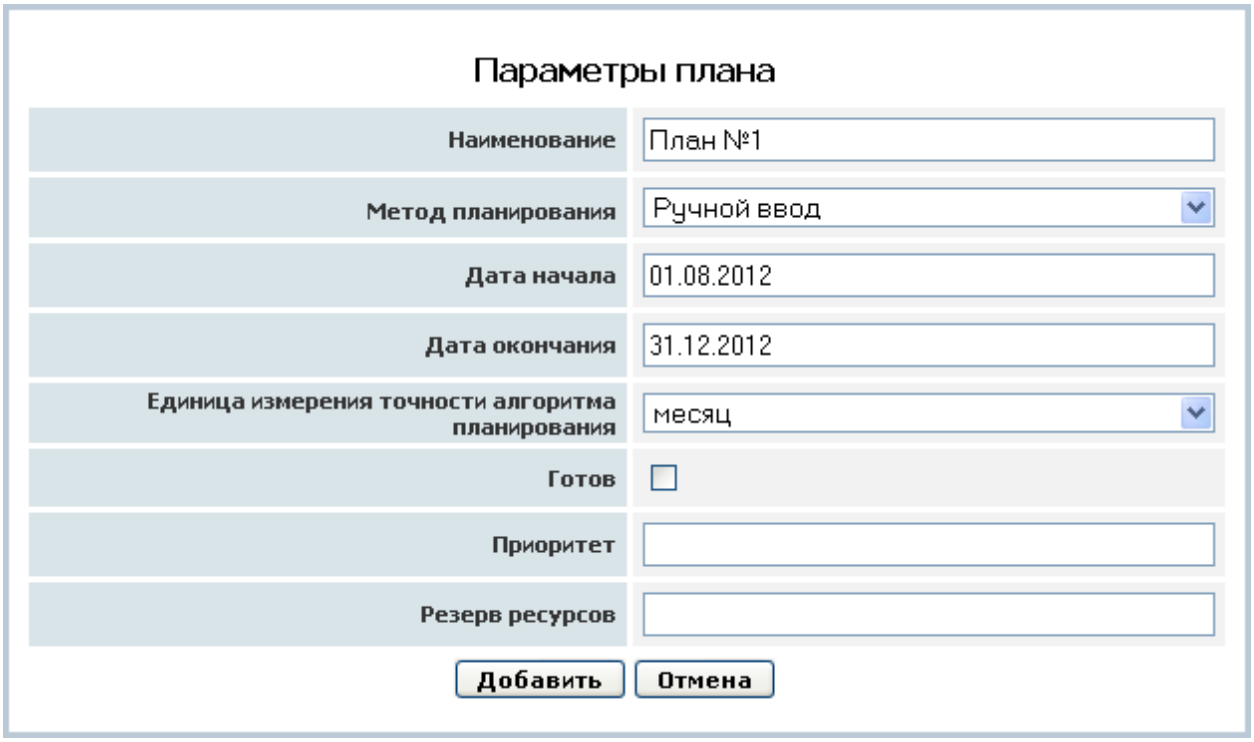


Рис. 5.28. Форма добавления плана.

После ввода параметров нужно нажать кнопку **Добавить**. В результате форма добавления плана исчезнет и новый план появится в списке планов. Для удаления плана нужно нажать кнопку в соответствующей строке.

Для редактирования параметров плана необходимо нажать на ссылку с именем ресурса из общего списка. При нажатии на ссылку откроется форма редактирования параметров плана аналогичная форме добавления плана. При изменении параметров плана, необходимо нажать кнопку **Сохранить**, чтобы изменения вступили в силу. В результате форма редактирования параметров плана исчезнет и новые параметры плана будут записаны в базу данных.

5.2.9. Запуск процедуры планирования и мониторинг результатов

Для запуска процедуры планирования следует выбрать в меню пункт **Планирование**. Планирование программ осуществляется всегда в рамках одной задачи. Задачу, с которой работает пользователь, можно изменить в настройках системы, перейдя туда по нажатию на кнопку **Настройки** в левом верхнем меню. Перед началом планирования оператор должен создать для каждой альтернативы не менее одной работы, которая осуществляется для реализации альтернативы. В созданных работах оператору необходимо ввести все ресурсы, затрачиваемые для каждой работы соответственно. Как создаются работы у альтернатив и вносятся ресурсы сказано в инструкции для операторов СППР. После создания работ и ввода ресурсов будет сформирована необходимая целостность данных, позволяющая перейти к планированию программ.

Для запуска процедуры планирования следует нажать кнопку . В результате будет сформирован план программы, который можно будет посмотреть нажав на кнопку **Диаграмма Ганта** в соответствующей строке. При нажатии на кнопку **Диаграмма Ганта** открывается новое окно, с отображением диаграммы Ганта, выбранного плана, рисунок 5.29.:

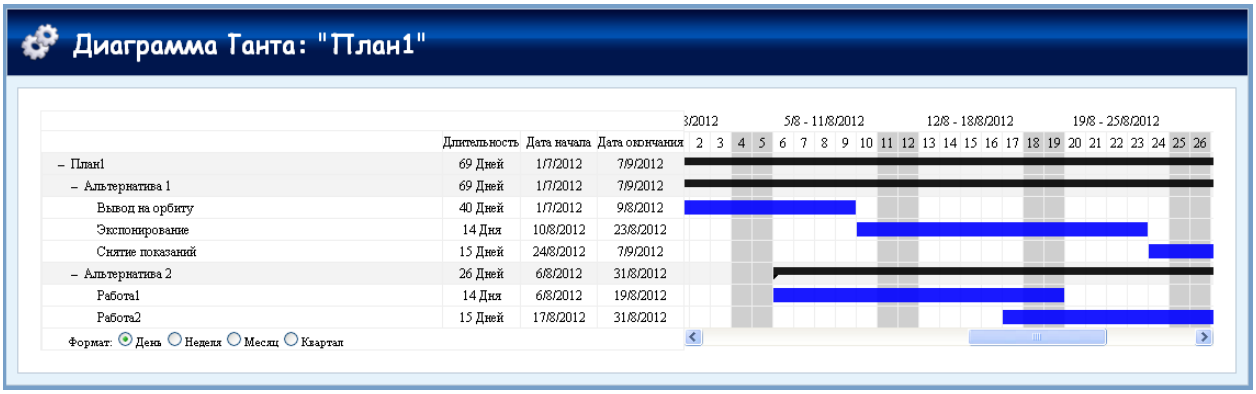


Рис. 5.29. Форма диаграммы Ганта выбранного плана.

Также для формирования плана программ в ручном режиме предусмотрен ручной ввод дат начала и окончания работ. Для ручного ввода дат необходимо нажать либо на наименование работы в левой части окна, либо на синий прямоугольник, показывающий графическое отображение работы на диаграмме. В результате откроется новое окно, в котором возможно изменить даты начала и окончания выбранной работы, рисунок 5.30.:

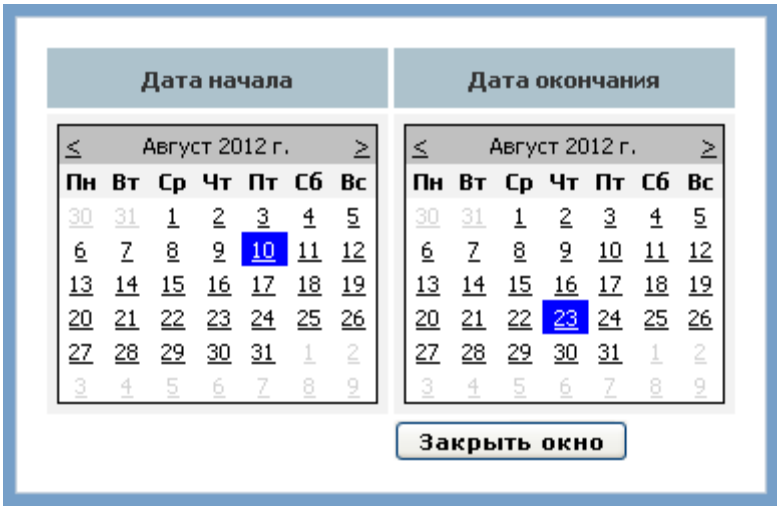


Рис. 5.30. Форма изменения сроков работ.

После изменения какой-либо даты в окне появляется кнопка **Сохранить**, по нажатию на которую данные записываются в базу данных. После изменения даты и сохранения результата, следует закрыть окно работы с датами, а затем обновить окно браузера с изображением диаграммы Ганта. В результате диаграмма Ганта текущего плана будет перерисована с учетом последних изменений.

5.3. Оценка эффекта от применения методологии

В таблице 5.2 показаны трудозатраты в чел.-мес. на разработку ряда специализированных СППР. Системы АСМО МО, АСКУ, СППР ГЛОНАСС создавались без применения унифицированной методологии. СППР «Космос» создана с применением унифицированной методологии.

Табл. 5.2

	Начальная стадия	Уточнение	Построение	Внедрение	Сумма чел-мес	Сумма чел-годы
АСМ МО	6	8	12	10	36	3,00
АСКУ	5	10	16	7	38	3,17
СППР ГЛОНАСС	7	9	14	7	37	3,08
СППР КОСМОС*	3	2	5	2	12	1,00

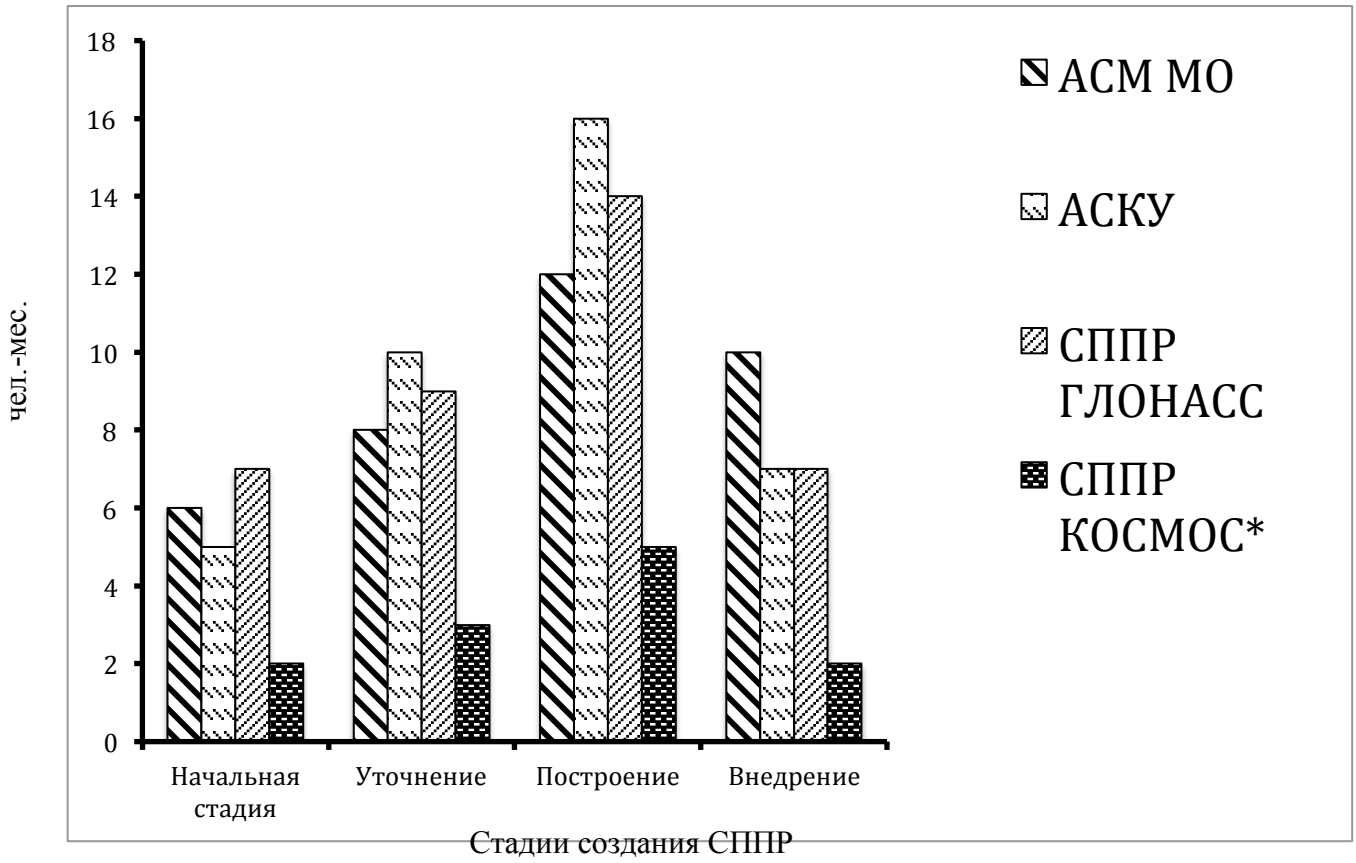


Рис. 5.30. Трудозатраты в разрезе стадий создания СППР

На рисунке 5.30 показаны графики трудозатрат в разрезе стадий разработки. Существенное улучшение на начальной стадии достигается за счет использования каркаса, который позволяет сразу переходить к формированию векторного критерия. Наибольшее сокращение сроков достигается на стадии уточнения. На данной фазе основное внимание уделяется проектированию и разработки структуры базы данных. унифицированная методология предлагает типовые решения данных вопросов. В целом, сроки разработки сокращаются в 3 раза.

5.4. Выводы по пятой главе

1. Использование принципов проектирования из методологии унифицированной разработки СППР, обеспечило быстрый (менее месяца) переход с фазы проектирования к фазе кодирования (написания ПО) СППР «Космос».
2. Каркасный подход позволил быстро (в течение одного дня) подключить к работающей СППР «Космос» метод мультипликативной свертки. Обновление затронуло только серверное ПО.
3. Использование унифицированной концептуальной модели БД СППР позволило гибко настраивать векторный критерий в процессе эксплуатации СППР «Космос»:

добавлять и удалять компоненты критерия, настраивать права доступа различным пользователям, оперативно редактировать систему ценностей ЛПР.

4. Трёхуровневая архитектура, предложенная в унифицированной методологии, позволила заказчику СППР «Космос» минимизировать затраты на сопровождение рабочих станций.
5. Реализация алгоритма оптимизации программ научно-прикладных исследований в рамках СППР «Космос», позволила получить рациональный план космических экспериментов соответствующий предпочтениям ЛПР.

Заключение

1. Разработан гибридный метод построения функций предпочтений, который содержит быструю итеративную процедуру ввода и редактирования предпочтений, обеспечивает требуемую точность поддержки принятия решений, позволяет учитывать зависимости по предпочтениям и позволяет работать с векторным критерием высокой размерности. Оригинальность метода заключается в процедуре разбиения пространства критериев на области и определения отношения доминирования между областями качественными методами теории принятия решений; а внутри области предложено применение количественных методов скаляризации векторного критерия.
2. Разработан метод дискретизации шкал критериев в иерархических структурах уменьшающий информационные потери, который позволяет строить дерево агрегирования показателей с большим количеством уровней при использовании функций предпочтений.
4. Создана методология унифицированной разработки СППР включающая: принципы проектирования СППР, унифицированную концептуальную модель БД, каркасный подход к трех уровневой архитектуре программного обеспечения СППР, набор рекомендаций по обеспечению качества программирования СППР. Методология унифицированной разработки СППР позволяет повысить скорость адаптации СППР под требования пользователя, снижает затраты на внедрение и сопровождение СППР, облегчает интеграцию СППР с другими информационными системами.
5. Методология унифицированной разработки СППР применена к задаче формирования программ космических экспериментов (КЭ) проводимых на российском сегменте Международной Космической Станции (МКС). На основе методологии была создана СППР «Космос». Она размещена в сети интернет по адресу http://knts.tsniimash.ru/DSS/. Открытые исходные тексты СППР «Космос» опубликованы на сайте https://code.google.com/p/microgravity/. С помощью гибридного метода была проведена оценка критерия «Значимость КЭ», информационные потери для критерия «Прикладной эффект КЭ» были снижены на 11%. Разработка СППР «Космос» заняла около года и методология унифицированной разработки сократила сроки примерно в 3 раза.

Список литературы

1. Абдрахимов Д., Иоффин А. Поддержка принятия решений: взгляд на место информационно-аналитических технологий ППР в арсенале банковского аналитика // Банковские технологии. 1997. – № 4. – С. 24-27.
2. Абрамов А.М., Капустин В.Ф. Математическое программирование – Л.: ЛГУ, 1976. – 328 с.
3. Авдуевский В.С, Осипов В.П., Шалимов. В.П. Создание информационно-аналитической системы для обеспечения исследований в области микрогравитации – насущная задача космического производства // Космонавтика и ракетостроение. 2000. – № 19. – С. 22-27
4. Агеев В.М., Запорожец А.В. Теория информации и кодирования. Дискретизация и кодирование измерений. – М.: Изд-во МАИ, 1977.
5. Адоладов К.Г., Бомас В.В. Автоматизированная система мониторинга факультета. // Современные технологии в задачах управления, автоматики и обработки информации: Сборник трудов XI международного научно-технического семинара. – М.: МГАПИ, 2002.
6. Айзерман М.А., Вольский В.И., Литваков Б.М. Элементы теории выбора. Псевдокритерии и псевдокритериальный выбор. – М.: 1994. – 293 с.
7. Анализ заявочной документации по предлагаемым и выполняемым на РС МКС космическим экспериментам с использованием компьютеризированной системы поддержки принятия решений по отдельным направлениям и группам космических исследований. – М.: НТО ИПМ. им. М.В. Келдыша РАН. – Шифр ОКР "МКС (Наука) - СППР-2010". – № Гос. регистрации Ф 40550. – Инв. №1012-1321/343-2010-1.
8. Анализ существующих методик ранжирования исследований и разработка новых методик с учетом комплексного анализа результатов исследований в 1975-2000 гг., достижений в развитии современных наземных технологий, актуальных направлений. – Королев: НТО ЦНИИМАШ. – Инв. № 851-5625/01-10 – 45- 175 /2001.
9. Артамонов Г.Т., Брехов М.О, Любатов Ю.В. и д.р. Теория проектирования вычислительных машин, систем и сетей. Под ред. Матова В.И. – М.: Изд-во МАИ, 1999.
10. Архангельский А.Я. Delphi 6. Справочное пособие. – М.: ЗАО «Издательство БИНОМ», 2001. – 1024 с.
11. Афонин К.А. Бомас В.В., Судаков В.А. Поддержка принятия многокритериальных решений по предпочтениям пользователя. СППР DSS/UTES. – М.: Изд-во МАИ, 2006. – 169 с.

12. Афонин К.А. Гибридная интеллектуальная система DSS/UTES // НТСб «Техника, Экономика», серия «Автоматизированные системы управления». – М.: ВНИИМИ, 1994.
13. Афонин К.А., Бомас В.В., Босин П.А. Формализация предпочтений пользователя на основе их непосредственного назначения //Методы и системы принятия решений. Автоматизация и интеллектуализация процессов проектирования и управления. – Рига: Рижский технический университет, 1993.
14. Афонин К.А., Бомас В.В., Ескин В.И. Система поиска рациональных решений, структурно инвариантная по отношению к проблемной области // Методы и системы принятия решений. Автоматизация и интеллектуализация процессов проектирования и управления. – Рига: Рижский технический университет, 1993.
15. Афонин К.А., Бомас В.В., Романов О.Т. Интеллектуальные системы в автоматизированном проектировании: учебное пособие. – М.: Изд-во МАИ, 1994. – 55 с.
16. Афонин К.А., Бомас В.В., Судаков В.А. Система поддержки принятия решений DSS/UTES [программа] // Программы для ЭВМ. Свидетельство № 990866 (03.12.1999).
17. Белкин А.Р., Левин М.Ш. Принятие решений: комбинаторные модели аппроксимации информации. – М.: Наука, 1990. – 160 с.
18. Белов В.С. ИНФОРМАЦИОННО-АНАЛИТИЧЕСКИЕ СИСТЕМЫ. Основы проектирования и применения: учебное пособие, руководство, практикум / Московский государственный университет экономики, статистики и информатики. – М.: 2005. – 111 с.
19. Бокс Д. Третья волна. // Решения Microsoft. 1998. – № 3. – С. 20-21.
20. Бомас В .В., Красовская М.А. Компьютерное исследование точности статистических методов: учебное пособие. – М.: Изд-во МАИ, 1998.
21. Бомас В.В. , Судаков В.А. Модель компонентных объектов в системе поддержки решений по предпочтениям пользователя // Сборник статей «Будущее авиации и космонавтики». – М.: Изд-во МАИ, 1999.
22. Бомас В.В., Ескин В.И., Судаков В.А. Модель компонентных объектов в системе поддержки решений по предпочтениям пользователя // Труды международного научно-технического семинара «Современные технологии в задачах управления, автоматики и обработки информации». – М.: Изд-во МАИ, 1999.
23. Бомас В.В., Ескин В.И., Судаков В.А. Оптимальное загрубление шкал в иерархии показателей // Современные технологии в задачах управления, автоматики и обработки информации: Сборник трудов XI международного научно-технического семинара. – М.: МГАПИ, 2002.

24. Бомас В.В., Ескин В.И., Судаков В.А. Система поддержки принятия решений-ГЛОНАСС [программа] // Программы для ЭВМ/ RU ОБПБТ № 3(80) 20.09.2012, Свидетельство № 2012613276 (06.04.2012).
25. Бомас В.В., Ленок А.И., Судаков В.А. Методическое и программное обеспечение автоматизации проверки достоверности данных формирующих программу космических экспериментов на РС МКС // Труды московской молодёжной научно-практической конференции «Инновации в авиации и космонавтике — 2012». – М.: 2012.
26. Бомас В.В., Леонов М.С. Сравнительный анализ современных методов поддержки многокритериальных решений // Авиакосмическое приборостроение. – М.: Научтехлитиздат, 2002. – №2.
27. Бомас В.В., Павленко А.И., Росин М.Ф. Оценка эффективности решений в АСУ ЛА по многим критериям. – М.: Изд-во МАИ,1998.
28. Бомас В.В., Судаков В.А. Поддержка субъективных решений в многокритериальных задачах. – М.: Изд-во МАИ, 2011. – 173 с.
29. Бомас В.В., Судаков В.А., Сурков В.В., Хахулин Г.Ф. Применение системы поддержки решений DSS/UTES в задачах мониторинга иерархических структур // Приборы и системы. Управление, контроль, диагностика. 2001. – № 9. – С. 70-73.
30. Бомас В.В., Сурков В.В., Судаков В.А. Автоматизированная система мониторинга муниципальных образований (АСМ МО) // Материалы семинара «Казначейская система исполнения бюджета и её модели». – М.: Изд-во «Элиот», 2002.
31. Бомас В.В., Сурков В.В., Судаков В.А. Система поддержки многокритериальных решений по предпочтениям пользователя (DSS/UTES) // Материалы семинара «Казначейская система исполнения бюджета и её модели». – М.: Изд-во «Элиот», 2002.
32. Борисов А.Н., Вилюмс Э.Р., Сукур Л.Я. Диалоговые системы принятия решений на базе мини-ЭВМ. – Рига: Зинатне, 1986. – 195 с.
33. Борисов А.Н., Левченков А.С. Методы интерактивной оценки решений. – Рига: Зинатне, 1982. – 250 с.
34. Брукс Ф. Мифический человеко-месяц или как создаются программные системы. – Пер. с англ. – СПб.: Символ-Плюс, 1999. – 304 с.
35. Бруно Дж. Л., Грэхем Р.Л., Коглер В.Г., Коффман Э.Г. мл., Сети Р., Ульман Дж.Д., Штиглиц К. Теория расписаний и вычислительные машины; под редакцией Э.Г. Коффмана. – М.: Наука, 1984. – 334 с.
36. Булыгин В.С., Красовская М.А., Хахулин Г.Ф. Теоретические основы автоматизированного управления (задачи, методы, алгоритмы теории оптимального планирования и управления). – М.: Изд-во МАИ, 2005. – 395 с.

37. Бусленко Н.П: Моделирование сложных систем. – М.: Наука, 1978. – 400 с.
38. Буч Г. Объектно-ориентированный анализ и проектирование с примерами приложений на C++, – 2-е изд. – М.: Изд-во «Бином», 1998. – 560 с.
39. Буч Г., Рамбо Д., Якобсон И. Введение в UML от создателей языка. 2-е изд.: Пер. с англ. Мухин Н. – М.: ДМК Пресс, 2012. – 496 с.
40. Вагнер Г. Основы исследования операций. – М: Изд-во «МИР», 1973. – 2 т. – 486 с.
41. Вендров А. М. Проектирование программного обеспечения экономических информационных систем: учебник. – М.: Финансы и статистика, 2002. – 352 с.
42. Вентцель Е.С. Теория вероятностей: учебник для студентов вузов. – М.: Издательский центр «Академия», 2003. – 576 с.
43. Вентцель Е.С., Овчаров Л.А. Теория вероятностей и ее инженерные приложения. – М.: Наука, 1988. – 480 с.
44. Венцель Е.С. Теория вероятностей: учебник для вузов. – 6-е изд. стер. – М.: Высшая школа, 1999. – 576 с.
45. Воронин А.Г., Лапин В.А., Широков А.Н. Основы управления муниципальным хозяйством. – М.: Дело, 1998. – 127 с.
46. Гаврилова Т.А., Хорошевский В.Ф. Базы знаний интеллектуальных систем. – СПб.: Питер, 2000. – 384 с.
47. Гафт М.Г., Подиновский В.В. О построении решающих правил в задачах принятия решений. - Автоматика и телемеханика, 1981. – № 6.
48. Гвоздева Т.В., Баллод Б.А. Проектирование информационных систем. – Ростов-на-Дону: Феникс, 2009. – 508 с.
49. Гибридные интеллектуальные системы, тезисы докладов Всесоюзной научно-практической конференции в 2-х частях, ВНИИПС, Научный совет АН СССР по проблеме "Искусственный интеллект", Ростовское областное управление ВНТО приборостроителей им. академика Вавилова, – Ростов-на-Дону, 1991.
50. Гласс Р. Факты и заблуждения профессионального программирования. – СПб.: Символ-Плюс, 2007. – 240 с.
51. Гончаров А.В., Григорьева А.А., Елкин К.С., Левтов В.Л.., Маров М.Я., Мухоян М.З., Осипов В.П., Сивакова Т.В., Шалимов В.П. Компьютерные технологии как средства поддержки принятия решений при выборе рациональных программ микрогравитационных исследований. Космонавтика и ракетостроение, 2006. – №4. – С. 105-112.
52. Горбунов-Посадов М.М. Расширяемые программы. – М.: Полиптих, 1999. – 336 с.

53. Горев А., Ахаян Р., Макашарипов С. Эффективная работа с СУБД. – СПб.: Питер, 1997. – 700 с.

54. ГОСТ Р 52017-2003. Аппараты космические. Порядок подготовки и проведения космического эксперимента. ИПК. Издательство стандартов, 2003.

55. Дегтярев Ю.И. Исследование операций. – М.: Высшая школа, 1986. – 320 с.

56. Дегтярев Ю.И. Формализованные методы исследования организационно-экономических задач как средство рационального расходования ограниченных ресурсов // Инноватика и экспертиза. Научные труды Федерального государственного бюджетного учреждения "Научно-исследовательский институт - Республиканский исследовательский научно-консультационный центр экспертизы (ФГБНУ НИИ РИНКЦЭ)". 2011. – № 1. – С. 60-65.

57. Долгосрочная программа научно-прикладных исследований и экспериментов, планируемых на российском сегменте МКС. Версия 2008 г. ЦНИИмаш. – Инв. №8004-31/1 от 28.11.08.

58. Ескин В.И., Судаков В.А. Гибридный метод формирования функций предпочтений в компьютеризированных системах поддержки принятий решений // Вестник Московского авиационного института. 2013. – № 3. – 20 т.– С. 166-172.

59. Ескин В.И., Судаков В.А. Автоматизированная поддержка решений с использованием гибридной функции предпочтений // Вестник МГТУ им. Н.Э. Баумана. Серия «Приборостроение». 2014. № 3. с. 116-124.

60. Инструкция по работе с программно-математическим обеспечением для лица принимающее решение (ЛПР). – Инв. № (105-1313-2011)-1313/140-2012-1.2.

61. Инструкция по работе с программно-математическим обеспечением для оператора. – Инв. № (105-1313-2011)-1313/140-2012-1.1.

62. Инструкция по работе с программно-математическим обеспечением для эксперта. – Инв. № (105-1313-2011)-1313/140-2012-1.3.

63. Калашников В.В. Организация моделирования сложных систем. – М.: Знание, 1982. – 64 с.

64. Калинин П.Е., Федосеев А.С., Судаков В.А. Формальная грамматика для построения детерминированных функций предпочтений // Современные технологии в задачах управления, автоматики и обработки информации: Сборник трудов XI международного научно-технического семинара. – М.: МГАПИ, 2002.

65. Карл И. Вигерс. Разработка требований к программному обеспечению/ Пер. с англ. – М.: Издательско-торговый дом «Русская Редакция», 2004. – 576 с.

66. Касчиато Ч. PowerBuilder. Руководство разработчика: Пер. с англ. – М.: Малип, 1997. – 390 с.

67. Кини Р.Л., Райфа Х. Принятие решений при многих критериях: предпочтения и замещения: Пер. с англ. / Под ред. И.Ф. Шахнова. – М.: Радио и связь, 1981. – 560 с.
68. Коды программно-математического обеспечения СППР «Космос 1.0». – Инв. № (105-1313-2011)-1313/140-2012-1.6.
69. Козлов В.Н. Системный анализ и принятие решений. – СПб.: Изд-во Политехнического университета, 2009. – 223 с.
70. Комягин В. Б. Программирование в Excel 5 и Excel 7 на языке Visual Basic. – М: Радио и связь, 1996. – 320 с.
71. Корн Г., Корн Т. Справочник по математике для научных работников и инженеров. – М.: Наука, 1973. – 832 с.
72. Корнеев В.В., Гареев А.Ф., Васютин С.В., Райх В.В. Базы данных. Интеллектуальная обработка информации. – М.: «Нолидж», 2000. – 352 с.
73. Кофман А., Анри-Лабордер А. Методы и модели исследования операций. Целочисленное программирование. – М.: Мир. 1977. – 432 с.
74. Крокет Ф. MFC. Мастерская разработчика. – М.: Издательско-торговый дом «Русская Редакция», 1998. – 440 с.
75. Круглински Д.Дж. Основы Visual C++. – М.: Издательско-торговый дом «Русская Редакция», 1997. – 830 с.
76. Ларичев О. И., Петровский А. В. Системы поддержки принятия решений. Современное состояние и перспективы их развития. // Итоги науки и техники. Сер. Техническая кибернетика. – М.: ВИНИТИ, 1987. – С. 131-164. – 21 т.
77. Ларичев О.И. Объективные модели и субъективные решения. – М.: Наука, 1987. – 144 с.
78. Ларичев О.И. Теория и методы принятия решений, а также Хроника событий в Волшебных странах: учебник. – М.: Логос, 2000. – 296 с.
79. Ларичев О.И., Мошкович Е.М. Качественные методы принятия решений. – М.: Наука, Физматлит, 1996. – 208 с.
80. Леоненков А.В. Самоучитель UML. – 2-е изд., перераб. и допол. – СПб.: БХВ-Петербург, 2006. – 472 с.
81. Липаев В.В. Качество программного обеспечения. – М.: Финансы и статистика, 1983. – 263 с.
82. Липаев В.В. Методы обеспечения качества крупномасштабных программных средств. – М.: Синтег, 2003. – 510 с.
83. Литвак Б.Г. Экспертная информация: Методы получения и анализа. – М: Радио и связь, 1982. – 184 с.

84. Литвак Б.Г. Государственное управление. Лучшие мировые практики. – М.: Синергия, 2012. - 224 с.

85. Лотов А.В., Поспелова И.И. Многокритериальные задачи принятия решений: учебное пособие. – М.: МАКС Пресс, 2008. – 197 с.

86. Львов В. Создание систем поддержки принятия решений на основе хранилищ данных. Системы Управления Базами Данных. 1997. – № 3. – С. 30-40.

87. М. Фаулер. UML. Основы. – 3-е изд., – пер. с англ. – СПб.: Символ-Плюс, 2006. – 192 с.

88. Майоров А.В. Москатов Г.К., Шибанов Г.П. Безопасность функционирования автоматизированных объектов. – М.: Машиностроение, 1988. – 264 с.

89. Макконнелл С. Совершенный код. Практическое руководство по разработке программного обеспечения. – СПб.: Питер, 2005. – 896 с.

90. Малышев В.В., Пиявский Б.С., Пиявский С.А. Метод принятия решений в условиях многообразия способов учета неопределенности, Известия РАН, серия «Теория и системы управления», 2010. – № 1. – С. 46-61.

91. Мартин Р.К. Ньюкирк Дж.В., Косс Р.С. Быстрая разработка программ. Принципы, примеры, практика. – Пер. с англ. А.П.Сергеева, Т.А. Шамренко. – М.: Вильямс, 2004. – 743 с.

92. Матвеев Л.А. Компьютерная поддержка решений: учебник. – СПб.: «Специальная Литература», 1998. – 472 с.

93. Международные стандарты, поддерживающие жизненный цикл программных средств. – М.: МП «Экономика», 1996. – 402 с.

94. Микони С.В. Многокритериальный выбор на конечном множестве альтернатив. – СПб.: Изд-во «Лань», 2009. – 272 с.

95. Миллер Д. А. Магическое число семь плюс или минус два. О некоторых пределах нашей способности перерабатывать информацию // Инженерная психология. – М.: Прогресс, 1964. – С. 192-225.

96. Монахов С.В. Событийно-ориентированная система имитационного моделирования для разработки дискретных, непрерывных и непрерывно-дискретных имитационных моделей: Диссертация на соискание ученой степени кандидата технических наук. – М.:, 2006.

97. Монахов С.В., Хахулин Г.Ф. Система имитационного моделирования Modelling для разработки дискретных и непрерывно-дискретных моделей. – М.: Изд-во МАИ, 2007. – 177 с.

98. Нечаев Ю.И., Осипов В.П. Концептуальный базис создания интегрированного вычислительного комплекса поддержки принятия решений // Нейрокомпьютеры: разработка, применение. 2011. – № 6. – С. 4-18.

99. Орлов А. И. Теория принятия решений: учебник для вузов. – М.: Экзамен, 2006. – 576 с.

100. Осипов В.П, Загреев Б.В., Судаков В.А. Система поддержки принятия решений для формирования программ исследований на МКС // Полет. Общероссийский научно-технический журнал. 2013. – № 10. – С. 28-41.

101. Осипов В.П., Репченков Р.О., Судаков В.А. Методическое и программное обеспечение автоматизации формирования программ космических экспериментов на РС МКС // IX Конференция молодых ученых «Фундаментальные и прикладные космические исследования». – М.: ИКИ РАН, 2012.

102. Осипов В.П., Репченков Р.О., Судаков В.А. Прикладная методика распределения ресурсов при планировании космических экспериментов // Материалы IX международной конференции по неравновесным процессам в соплах и струях (NPNJ’2012). – Алушта: 2012.

103. Осипов В.П., Сивакова Т.В., Судаков В.А. Предпосылки унификации программных средств поддержки принятия решений // Программные продукты и системы. 2013. – №3. – С. 147-150.

104. Осипов В.П., Сивакова Т.В., Судаков В.А., Загреев Б.В., Трахтенгерц Э.А. Методологические основы поддержки принятия решений при планировании научно-прикладных исследований и экспериментов на международной космической станции (МКС) // Электротехнические и информационные комплексы и системы. 2013. – № 3. – 9 т. – С. 80-88.

105. Осипов В.П., Судаков В.А. О задаче рационального распределения ресурсов при планировании научно-прикладных исследований на российском сегменте МКС // Материалы XVIII Международной конференции по Вычислительной механике и современным прикладным программным системам (ВМСППС’2013). – Алушта: 2013.

106. Осипов В.П., Судаков В.А. Принципы унификации архитектуры систем поддержки принятия решений // Материалы XVIII Международной конференции по Вычислительной механике и современным прикладным программным системам (ВМСППС’2013). – Алушта: 2013.

107. Осипов В.П., Судаков В.А., Хахулин Г.Ф. Информационные технологии формирования этапной программы научно-прикладных исследований на российском сегменте Международной космической станции // Вестник компьютерных и информационных технологий. 2012. – №12. – С. 24-28.

108. Осуга С. Обработка знаний: Пер. с япон. – М.: Мир, 1989. – 293 с.
109. Осуга С., Саэки Ю. Приобретение знаний: Пер. с япон. – М.: Мир, 1990. – 304 с.
110. П.В. Горский Положение об аналитическом рейтинге рангового типа http://www.gorskiy.ru/Articles/ratrul.html.
111. Пападимитриу Х., Стайглиц К. Комбинаторная оптимизация. Алгоритмы и сложность. – М.: Мир, 1985. – 510 с.
112. Петрова Б.Н. Вопросы управления космическими аппаратами. – М.: Мир, 1975. – 217 с.
113. Петровский А.Б. Теория принятия решений. – М.: Издательский центр «Академия», 2009. – 400 с.
114. Пиявский С.А. Технология ПРИНН для моделирования слабо структурированных систем // Информационные технологии в моделировании и управлении: Труды II Международной научно-пректической конференции. – СПб.: 2000.
115. Подиновский В. В. Введение в теорию важности критериев в многокритериальных задачах принятия решений. – М.: Физматлит, 2007. – 64 с.
116. Подиновский В.В. Многокритериальные задачи с упорядоченными по важности критериями // Автоматика и телемеханика. 1976. – №11.
117. Подиновская О. В., Подиновский В. В. Информация о важности групп критериев в многокритериальных задачах принятия решений. I. Качественная информация. Равноважные группы критериев равной важности // Информационные технологии моделирования и управления. 2014. № 1. С. 58-67.
118. Поспелов Д.А. Ситуационное управление: Теория и практика. – М.: Наука, 1986. – 288 с.
119. Программно-математическое обеспечение СППР «Космос 1.0». – Инв. № (105-1313-2011)-1313/140-2012-1.5.
120. Пфанцагль И. Теория измерений. – М: 1976. – 248 с.
121. Р. Т. Фатрелл, Д. Ф. Шафер, Л. И. Шафер. Управление программными проектами. Достижение оптимального качества при минимуме затрат.: Пер. с англ. – М.: Вильямс, 2003. – 1125 с.
122. Разработка и развитие существующих прикладных информационных систем для поддержки деятельности местного самоуправления. Технический отчет по теме: «Автоматизированная система мониторинга муниципальных образований». Руководитель Бомас В.В. – М.: ЗАО «ОВИОНТ», 2001.
123. Разработка информационно-аналитической системы подготовки и принятия решений по направлению «космическая технология» // – М.: НТО ИПМ им. М.В. Келдыша РАН. – Шифр НИР "СППР-1". 2006. – Инв. № 0602-1045/111-06-01.

124. Разработка концепции информационной поддержки российских ученых и специалистов, принимающих участие в планировании и проведении микрогравитационных исследований // – М.: НТО ИПМ. им. М.В. Келдыша РАН. – Шифр НИР «СППР-1». – Инв. № 0602-1045/267-08-02.

125. Разработка концепции экспертной оценки микрогравитационных исследований. // – М.: НТО ИПМ им. М.В. Келдыша РАН. – Шифр НИР "СППР", КНИР "ГРАНТ". 2004. – Инв. № 0428-4507/278-04-01.

126. Разработка метода и программно-математического обеспечения (ПМО) оптимизации планирования исследований и экспериментов для формирования Этапной программы НПИ. – М.: НТО ИПМ им. М.В. Келдыша РАН, – Шифр ОКР «Наука–ИПМ». – Инв. № (105-1313-2011)-1313/140-2012-1.

127. Разработка метода и программно-математического обеспечения определения приоритетности и оптимизации планирования исследований и экспериментов для формирования Этапной программы НПИ. Выполнение работ постановщика по подготовке, реализации КЭ «Дальность» на борту РС МКС в 2011 и 2012 годах, обработке и анализу полученных результатов». – М.: НТО ИПМ им. М.В. Келдыша РАН. – Шифр ОКР «Наука–ИПМ». – Инв. № (105-1313-2011)-1313/175-2011-1.

128. Разработка пакета программного обеспечения внешних ресурсов и удаленного доступа к ним для информационной поддержки российских ученых и специалистов, принимающих участие в планировании и проведении микрогравитационных исследований. – М.: НТО ИПМ. им. М.В. Келдыша РАН. – Шифр НИР «СППР-1». – Инв. № 0901-1323.133-2009-1.

129. Разработка программных комплексов оценки состояния сил и средств и поддержки управляющих решений. Технический отчет по теме: «Автоматизированная система контроля состояний и поддержки управляющих решений». Руководитель Хахулин Г.Ф. – М.: Изд-во МАИ, 2001.

130. Росин М.Ф. Принципы и модели принятия решений в АСУ ЛА. – М.: Изд-во МАИ,1994.

131. Росин М.Ф., Булыгин В.С. Статистическая динамика и теория эффективности систем управления: учебник для вузов. – М.: Машиностроение, 1981. – 312 с.

132. Российский сегмент МКС. Справочник пользователя // Ракетно-космическая корпорация «Энергия» имени С. П. Королева. 2012.

133. С.В. Черемных, И.О. Семенов, В.С. Ручкин. Структурный анализ систем: IDIF-технологии. – М.: Финансы и статистика, 2003. – 208 с.

134. Саати Т. Принятие решений. Метод анализа иерархий. – М.: Радио и Связь, 1993. – 278 с.

135. Садофьев М.В., Судаков В.А. Разработка Системы Поддержки Решений DSS/UTES на основе идеологии Открытого Программного Обеспечения // Современные технологии в

задачах управления, автоматики и обработки информации: Труды XVII Международного научно-технического семинара. Алушта, 2008 г. – СПб.: ГУАП, 2008.

136. Саукап Р. Основы Microsoft SQL Server 6.5. / Пер. с англ. – М.: Издательско-торговый дом «Русская Редакция», 1999.
137. Семухин М.В. Судаков В.А. Геокодирование в автоматизированных системах маршрутизации транспорта // Современные технологии в задачах управления, автоматики и обработки информации. Международный научно-технический семинар. – Алушта, 2011.
138. Соболь И.М., Статников Р.Б. Выбор оптимальных параметров в задачах со многими критериями. – М.: Дрофа, 2006. – 175 с.
139. Советов Б.Я., Яковлев С.А. Моделирование систем: учебник для вузов. – М.: Высшая школа, 1985. – 319 с.
140. Создание и тестирование программного обеспечения СППР для технологии «клиент-сервер» – М.: НТО ИПМ им. М.В. Келдыша. – Шифр НИР "СППР-1". 2007. – Инв. № 0602-1045/111-07-01.
141. Сокуренко Е.А., Хахулин Г.Ф. Основы моделирования АСУ. – М.: Изд-во МАИ, 1990. – 60 с.
142. Судаков В.А. Автоматизация процесса управления разработкой корпоративной информационной системы // Вестник Московского авиационного института. 2010. – № 1. – 17 т. – С. 149-153.
143. Судаков В.А. Инструментальные средства для управления разработкой корпоративной информационной системы (окончание) // Вестник компьютерных и информационных технологий. 2008. – № 11. – С. 51-55.
144. Судаков В.А. Инструментальные средства для управления разработкой корпоративной информационной системы // Вестник компьютерных и информационных технологий. 2008. – № 10. – С. 49-55.
145. Судаков В.А. Мониторинг иерархических структур // Материалы 8-ой Всероссийской межвузовской научно-технической конференции студентов и аспирантов «Микроэлектроника и информатика - 2001». –М.: МИЭТ, 2001.
146. Судаков В.А. Обратное проектирование и оценка качества программных систем. – М.: Изд-во "Полиграф Центр", 2007. 38 с.
147. Судаков В.А. Оценка качества процессов разработки корпоративной информационной системы // Современные технологии в задачах управления, автоматики и обработки информации: Труды XVII Международного научно-технического семинара. Алушта, сентябрь 2008 г. – СПб.: ГУАП, 2008.

148. Судаков В.А. Решение многокритериальных задач высокой размерности в системе поддержки решений DSS/UTES // Материалы V Всероссийской конференции студентов и аспирантов «Техническая кибернетика, радиоэлектроника и системы управления». – Таганрог: 2000.

149. Судаков В.А. Система поддержки решений ориентированная на предпочтения пользователя // Материалы семинара по искусственному интеллекту. – М.: НПК «ПОИСК», 2000.

150. Суздальцев А.П., Судаков В.А. Реализация гибридного метода выявления предпочтений в задаче ранжирования космических экспериментов проводимых на российском сегменте МКС // 12-я Международная конференция «Авиация и Космонавтика – 2013», 12-15 ноября 2013 года. Москва. Тезисы докладов. – СПб.: Мастерская печати, 2013. – С. 510-512.

151. Трахтенгерц Э.А. Компьютерная поддержка принятия решений. – М.: СИНТЕГ, 1998. – 376 с.

152. Трахтенгерц Э.А. Компьютерные методы реализации экономических и информационных управленческих решений. В 2-х томах. // Методы и средства. – М.: СИНТЕГ, 2009. – 1 т. – 172 с. // Реализация решений. - М.: СИНТЕГ, 2009. – 2 т. – 224 с.

153. Трепалин С. Сервер автоматизации в Delphi. // Компьютер пресс. 1998. – № 1. – С. 132-134.

154. Урличич Ю.М., Данилин Н.С. Управление качеством космической радиоэлектронной аппаратуры в условиях глобальной открытой экономики. – М.: МАКС Пресс, 2003. – 204 с.

155. Федотов А.В. Программирование задач моделирования и оптимизации на языке Object Pascal: учебное пособие. – М.: Изд-во МАИ, 2001. – 112 с.

156. Хахулин Г.Ф. Основы конструирования имитационных моделей. Учебное пособие. 2-е изд. доп. и исправ. – М.: НПК «Поток», 2002. – 221 с.

157. Хахулин Г.Ф. Постановки и методы решения задач дискретного программирования. – М.: Изд-во МАИ, 1992. – 59 с.

158. Цыгичко В.Н. Руководителю о принятии решений. – М.: Инфра-М, 1996. – 272 с.

159. Чэппел Д., Технологии ActiveX и OLE / Пер. с англ. – М.: Издательско-торговый дом «Русская Редакция», 1997. – 320 с.

160. Шапот М. Интеллектуальный анализ данных в системах поддержки принятия решений // Открытые системы. 1998. – № 1. – С. 30-35.

161. Шишкин Е.В., Чхартишвили А.Г. Математические методы и модели в управлении: учеб. пособие. – М.: Дело, 2000. – 440 с.

162. Щербинин В.В., Кветкин Г.А., Свиязов А.В., Андриенко В.Б. Навигационное обеспечение системы посадки беспилотных летательных аппаратов // Гироскопия и навигация. 2013. – № 1 (80). – С 19-33.

163. Якобсон А., Буч Г., Рамбо Дж. Унифицированный процесс разработки программного обеспечения.: Пер. с англ. – СПб.: Питер, 2002. – 496 с.

164. Aphonin K.A., Bomas V.V., Eskin V.I. Use of the Hybrid Intellectual System with Situational Knowledge Base for Optimizing of Parameters of Objects and Control Systems // Proceedings of the 1993 MAI/BUAA International Symposium on Automatic Control. —Moscow, 1993. — Volume 1, Part 1.

165. B.V. Zagreev, V.P. Osipov, V.A. Sudakov. A Decision Support System (DSS) for Developing Programs of Scientific and Applied Research and Experiments on the Russian Segment of the ISS // 5th European Conference for Aeronautics and Space Sciences (EUCASS 2013). Munich, Germany, 1-4 July 2013.

166. Boje D.M., Murningham J.K. Group confidence pressures in iterative decision // Manag. Sci. 1982, V. 28, №10, p. 1187-1196.

167. Charnes A., Cooper W.W., Deterministic equivalents for optimizing and satisfying under chance constraints, Oper. Res., 11,1. 1963.

168. Dalxey N., Helmer O. Experimental application of the Delphi method to the use of experts // Manag. Sci, V. 9, № 3, p. 458-467, 1963.

169. Dempster M.A.H. On stohastic programming, J. Math. Anal, and Appl., 21,2. 1968.

170. Dongarra, J.J., et al. 1979, LINPACK User's Guide (Philadelphia: S.I.A.M.).

171. Dyer R.F., Forman E. A., Forman E. H., Jouflas G., CASE STUDIES IN MARKETING DECISIONS USING EXPERT CHOICE, Pittsburgh, Expert Choice, Inc. Publ., 1988.

172. Eom S.B. Decision support systems research: reference disciplines and a cumulative tradition. - The International Journal of Management Science, 23, 5, October 1995, p. 511-523.

173. French S. The role of sensitiving analysis in decision analysis in decision analysis. In Executive Information Systems and Deciion Making. Chapman Hall, London, 1991.

174. Ginzberg M.J., Stohr E. A. A decision support: Issues and Perspectives. - Processes and Tools for Decision Support. Amsterdam, North - Holland Publ. Co, 1983.

175. Golub, G.H., and Van Loan, C.F. 1989, Matrix Computations, 2nd ed.(Baltimore: Johns Hopkins University Press).

176. Hamdy A. Taha. Operations Research: An Introduction: International Edition, 8/E 2007.

177. ISO/IEC 9126-1:2001. Software engineering — Software product quality — Part 1: Quality model.

178. J. McCall, P. Richards, G. Walters. Factors in Software Quality. three volumes, NTIS AD-A049-014, AD-A049-015, AD-A049-055, November 1977.

179. Raiffa H. Decision Analysis. Introductory lectures on choices under Uncertainty. Addison Wesley, 1968.

180. Rios Insua D. Sensetivity Analysis in Multy-Objective Decision Making. Lecture Notes in Economics and Mathematical Systems N 347. Springer Verlag, Berlin, 1990.

181. Roy B. Methodologie Multicritere d'Aide a la Decision. Economica, 1985.

182. Simon H.A. How big is a chunk // Science. №183. 1974.

183. Thomsen E. Decision Alchemy. // Decision Support. Intelligent Enterprise Volume 1 - № 1, 1998.

184. Tversky A. Intransitivity of preferences // Psychological Review. № 76. 1969.

185. Vicki L. Sauter, Decision Support Systems for Business Intelligence, 2nd Edition, John Wiley & Sons, Inc., Hoboken, New Jersey, 2011.

186. Wakker P. Additive Representations of Preferences: a New Foundation of Decision Analysis. Kluwer, Dordrecht, 1989.

187. Walson S.R., Buede D. Decision Synthesis: the Principles and Practice of Decision Analysis Combridge University Press, Cambridge, 1987.

Приложения

Приложение 1. Трассировка гибридного метода

Трассировка решения задачи оценки значимости КЭ:

```
[1]  ------------------------------------------------------------
[2]  Исходные данные в формате JSON
[3]  {
[4]  "criterions": [
[5]  {"name":"Научный эффект КЭ", "scale": ["н/у",  "неуд",  "уд",  "хор", "отл"], "rank": 0.19697},
[6]  {"name":"Прикладной эффект КЭ", "scale": [98,  155, 173, 256], "rank": 0.19697, "min":0},
[7]  {"name":"Международное участие", "scale": ["н/у",  "нет",  "да"], "rank": 0.093939},
[8]  {"name":"Технологическая инновационность", "scale": [0.2,  0.4,  1], "rank": 0.157576, "min":0},
[9]  {"name":"Актуальность КЭ", "scale": ["н/у",  "неуд",  "уд",  "хор", "отл"], "rank": 0.19697},
[10] {"name":"Область применения результатов КЭ", "scale": ["неуд",  "уд",  "хор", "отл"], "rank": 0.157576}
[11] ],
[12] "pref_scale": ["неуд", "уд", "хор", "отл"],
[13] "preference": {
[14] "неуд":[
[15] {"Научный эффект КЭ":["н/у",  "неуд"]},
[16] {"Прикладной эффект КЭ":[98]},
[17] {"Актуальность КЭ":  ["н/у", "неуд"]}
[18] ],
[19] "уд":[
[20] {"Научный эффект КЭ":  ["уд"], "Прикладной эффект КЭ":  [155],  "Технологическая инновационность":  [0.2],
[21] "Актуальность КЭ":  ["уд"], "Область применения результатов КЭ":  ["уд"], "Международное участие": ["нет"]}
[22] ],
[23] "хор":[
[24] {"Научный эффект КЭ":  ["уд"], "Прикладной эффект КЭ":  [155],  "Технологическая
```

инновационность": [0.2],
[25] "Актуальность КЭ": ["уд"], "Область применения результатов КЭ": ["уд"], "Международное участие": ["да"]},
[26] {"Научный эффект КЭ": ["уд"], "Прикладной эффект КЭ": [173], "Технологическая инновационность": [0.2],
[27] "Актуальность КЭ": ["уд"], "Область применения результатов КЭ": ["уд"], "Международное участие": ["нет"]}
[28]],
[29] "отл":[
[30] {"Научный эффект КЭ": ["уд"], "Прикладной эффект КЭ": [256], "Технологическая инновационность": [0.4],
[31] "Актуальность КЭ": ["хор"], "Область применения результатов КЭ": ["отл"]},
[32] {"Научный эффект КЭ": ["отл"], "Прикладной эффект КЭ": [173], "Технологическая инновационность": [0.2],
[33] "Актуальность КЭ": ["отл"], "Область применения результатов КЭ": ["хор"]}
[34]]
[35] },
[36] "alternative": {
[37] "Акустика-М":[2,217,2,0.37,2,2],
[38] "Альбедо":[4,176,2,1,4,2],
[39] "АРИЛ":[4,160,2,0.32,3,2],
[40] "Астра-3":[3,187,2,0.07,4,2],
[41] "Астровакцина":[2,186,2,0.07,3,2],
[42] "Бактериофаг":[4,201,1,0.25,4,2],
[43] "Биоген":[4,104,1,0.16,3,3],
[44] "Биодеградация":[2,163,2,0.93,2,2],
[45] "Биомагнистат":[2,102,2,0.72,2,2],
[46] "Биотест":[0,59,0,0.61,0,1],
[47] "Биотрек":[4,256,2,0.68,3,2],
[48] "Биоэмульсия":[2,187,2,0.61,4,2],
[49] "БИФ":[4,142,2,0.45,3,2],
[50] "БСМК":[3,177,1,0.74,3,2],
[51] "БТН-Нейтрон-2":[0,59,0,0.45,0,2],
[52] "Бюон":[3,59,0,0.93,0,2],
[53] "Вектор-Т":[2,30,0,0.06,3,2],

```
[54]  "Великое начало":[1,143,2,0.3,1,2],
[55]  "Ветерок":[2,187,1,0.13,2,2],
[56]  "Вибролаб":[4,116,2,0.26,3,2],
[57]  "Визир":[2,155,2,0.73,2,1],
[58]  "ВИРУ":[2,155,2,0.81,2,2],
[59]  "Виток-2":[0,59,0,0.85,0,1],
[60]  "Выносливость":[4,30,0,0.77,3,2],
[61]  "Гематология":[0,74,0,0.3,0,2],
[62]  "ГРИС – ФКИ – 1":[2,98,1,0.37,3,2],
[63]  "Дальность":[2,158,2,0.11,2,2],
[64]  "Диурез":[0,59,0,0.7,0,1],
[65]  "Диффузия":[2,143,2,0.61,2,2],
[66]  "Дыхание":[0,59,0,0.34,0,1],
[67]  "Женьшень-2":[4,160,2,0.98,3,2],
[68]  "Захват-Э":[3,202,2,0.14,3,2],
[69]  "Идентификация":[4,201,2,0.51,4,3],
[70]  "Изгиб":[4,201,2,0.43,3,3],
[71]  "Изменчивость":[4,199,2,0.91,3,0],
[72]  "Инфразвук-М":[2,143,2,0.48,4,2],
[73]  "Искажение":[2,143,2,0.26,3,2],
[74]  "Кальций":[4,143,2,0.2,2,3],
[75]  "Капля-2":[0,45,0,0.72,0,2],
[76]  "КЛПВЭ":[4,127,2,0.48,3,3],
[77]  "Кожа":[4,201,2,0.65,3,0],
[78]  "Контур":[3,200,2,0.33,2,2],
[79]  "Контур-2":[3,230,2,0.96,3,2],
[80]  "Конъюгация":[4,172,2,0.13,4,2],
[81]  "Криоконсервация":[4,177,1,0.15,3,2],
[82]  "Кромка":[2,215,2,0.35,4,2],
[83]  "Кулоновский кристалл":[2,158,2,0.68,2,2],
[84]  "Лактолен":[4,160,2,0.57,3,2],
[85]  "МАИ-75":[2,173,2,0.19,2,0],
[86]  "МАТИ-75":[1,143,2,0,2,1],
[87]  "Матрешка-Р":[4,116,2,0.58,2,1],
[88]  "МВН":[3,142,2,0.59,3,3],
```

[89] "Межклеточное взаимодействие":[0,59,0,0.81,0,2],
[90] "Мембрана":[3,157,2,0.96,3,3],
[91] "Метаболизм":[4,117,2,0.6,2,0],
[92] "МСК":[3,177,1,0.27,3,2],
[93] "Наноспутник":[3,113,0,0.32,3,2],
[94] "ОЧБ":[4,160,2,0.15,3,2],
[95] "Перспектива-КМ":[4,116,2,0.36,3,2],
[96] "Пилот":[0,74,0,0.1,0,2],
[97] "Плазма-ЭРП":[2,128,1,0.12,2,2],
[98] "Платан":[3,173,2,0.23,3,2],
[99] "Пневмокард":[0,74,0,0.63,0,2],
[100] "Полиген":[0,59,0,0.85,0,2],
[101] "Привязка":[2,173,2,0.97,2,2],
[102] "Пульс":[0,74,0,0.45,0,2],
[103] "РадиоСкаф":[3,158,2,0.4,2,1],
[104] "Регенерация":[0,59,0,0.01,0,2],
[105] "Реставрация":[0,59,0,0.51,3,1],
[106] "Рецептор":[0,59,0,0.45,0,2],
[107] "Сейнер":[2,59,0,0.64,0,3],
[108] "Секция ЭДК":[4,173,2,0.84,4,2],
[109] "Сепарация":[3,187,2,0.1,2,2],
[110] "Скорпион":[2,158,2,0.15,3,2],
[111] "СЛС":[4,175,2,0.24,3,3],
[112] "Сонокард":[0,74,0,0.51,0,2],
[113] "Среда МКС":[2,173,2,0.96,2,2],
[114] "Статокония":[0,59,0,0.69,0,2],
[115] "Стресс":[2,133,2,0.58,3,1],
[116] "Структура":[4,172,2,0.1,3,2],
[117] "Тензор":[4,173,2,0.89,3,2],
[118] "Тень-Маяк":[2,155,2,0.12,2,2],
[119] "Типология":[0,74,0,0.15,0,2],
[120] "Токсичность":[3,118,0,0.16,3,1],
[121] "Трибокосмос":[4,158,2,0.77,3,2],
[122] "Трос-МГТУ":[0,74,0,0.84,0,1],
[123] "Фазопереход":[4,199,2,0.78,4,2],

```
[124]  "Фермент":[3,157,2,0.31,3,3],
[125]  "Физика-Образование":[2,143,2,0.3,2,2],
[126]  "Химия-Образование":[2,158,2,0.22,2,1],
[127]  "Эколинс":[2,214,2,0.47,3,3],
[128]  "Экология-Образование":[2,158,2,0.17,2,2],
[129]  "Эпсилон-НЭП":[4,148,2,0.52,3,3]
[130]  }
[131]  }
[132]
[133]  ------------------------------------------------------------
[134]  Заполнение структур данных и проверка на пересечение множеств
[135]
[136]  Множество № 0 Уровень предпочтений P = 0
[137]  [[0, 1], [0, 1, 2, 3], [0, 1, 2], [0, 1, 2], [0, 1, 2, 3, 4], [0, 1, 2, 3]]
[138]  пересечений не найдено
[139]  Множество № 1 Уровень предпочтений P = 0
[140]  [[0, 1, 2, 3, 4], [0], [0, 1, 2], [0, 1, 2], [0, 1, 2, 3, 4], [0, 1, 2, 3]]
[141]  пересечений не найдено
[142]  Множество № 2 Уровень предпочтений P = 0
[143]  [[0, 1, 2, 3, 4], [0, 1, 2, 3], [0, 1, 2], [0, 1, 2], [0, 1], [0, 1, 2, 3]]
[144]  пересечений не найдено
[145]  Множество № 3 Уровень предпочтений P = 1
[146]  [[2], [1], [1], [0], [2], [1]]
[147]  пересечений не найдено
[148]  Множество № 4 Уровень предпочтений P = 2
[149]  [[2], [1], [2], [0], [2], [1]]
[150]  пересечений не найдено
[151]  Множество № 5 Уровень предпочтений P = 2
[152]  [[2], [2], [1], [0], [2], [1]]
[153]  пересечений не найдено
[154]  Множество № 6 Уровень предпочтений P = 3
[155]  [[2], [3], [0, 1, 2], [1], [3], [3]]
[156]  пересечений не найдено
[157]  Множество № 7 Уровень предпочтений P = 3
[158]  [[4], [2], [0, 1, 2], [0], [4], [2]]
```

[159] пересечений не найдено

[160]

[161] ----------

[162] Поиск ячеек в графе

[163]

[164] q = [5, 4, 3, 3, 5, 4] Q = 3600

[165] Добавлена ячейка T[0] = [2, 1, 0, 0, 2, 0]

[166] Добавлена ячейка T[1] = [3, 1, 0, 0, 2, 0]

[167] Добавлена ячейка T[2] = [4, 1, 0, 0, 2, 0]

[168] Добавлена ячейка T[3] = [2, 2, 0, 0, 2, 0]

[169] Добавлена ячейка T[4] = [3, 2, 0, 0, 2, 0]

[170] Добавлена ячейка T[5] = [4, 2, 0, 0, 2, 0]

[171] Добавлена ячейка T[6] = [2, 3, 0, 0, 2, 0]

[172] Добавлена ячейка T[7] = [3, 3, 0, 0, 2, 0]

[173] Добавлена ячейка T[8] = [4, 3, 0, 0, 2, 0]

[174] Добавлена ячейка T[9] = [2, 1, 1, 0, 2, 0]

[175] Добавлена ячейка T[10] = [3, 1, 1, 0, 2, 0]

[176] Добавлена ячейка T[11] = [4, 1, 1, 0, 2, 0]

[177] Добавлена ячейка T[12] = [2, 2, 1, 0, 2, 0]

[178] Добавлена ячейка T[13] = [3, 2, 1, 0, 2, 0]

[179] Добавлена ячейка T[14] = [4, 2, 1, 0, 2, 0]

[180] Добавлена ячейка T[15] = [2, 3, 1, 0, 2, 0]

[181] Добавлена ячейка T[16] = [3, 3, 1, 0, 2, 0]

[182] Добавлена ячейка T[17] = [4, 3, 1, 0, 2, 0]

[183] Добавлена ячейка T[18] = [2, 1, 2, 0, 2, 0]

[184] Добавлена ячейка T[19] = [3, 1, 2, 0, 2, 0]

[185] Добавлена ячейка T[20] = [4, 1, 2, 0, 2, 0]

[186] Добавлена ячейка T[21] = [2, 2, 2, 0, 2, 0]

[187] Добавлена ячейка T[22] = [3, 2, 2, 0, 2, 0]

[188] Добавлена ячейка T[23] = [4, 2, 2, 0, 2, 0]

[189] Добавлена ячейка T[24] = [2, 3, 2, 0, 2, 0]

[190] Добавлена ячейка T[25] = [3, 3, 2, 0, 2, 0]

[191] Добавлена ячейка T[26] = [4, 3, 2, 0, 2, 0]

[192] Добавлена ячейка T[27] = [2, 1, 0, 1, 2, 0]

[193] Добавлена ячейка T[28] = [3, 1, 0, 1, 2, 0]

[194] Добавлена ячейка T[29] = [4, 1, 0, 1, 2, 0]

[195] Добавлена ячейка T[30] = [2, 2, 0, 1, 2, 0]

[196] Добавлена ячейка T[31] = [3, 2, 0, 1, 2, 0]

[197] Добавлена ячейка T[32] = [4, 2, 0, 1, 2, 0]

[198] Добавлена ячейка T[33] = [2, 3, 0, 1, 2, 0]

[199] Добавлена ячейка T[34] = [3, 3, 0, 1, 2, 0]

[200] Добавлена ячейка T[35] = [4, 3, 0, 1, 2, 0]

[201] Добавлена ячейка T[36] = [2, 1, 1, 1, 2, 0]

[202] Добавлена ячейка T[37] = [3, 1, 1, 1, 2, 0]

[203] Добавлена ячейка T[38] = [4, 1, 1, 1, 2, 0]

[204] Добавлена ячейка T[39] = [2, 2, 1, 1, 2, 0]

[205] Добавлена ячейка T[40] = [3, 2, 1, 1, 2, 0]

[206] Добавлена ячейка T[41] = [4, 2, 1, 1, 2, 0]

[207] Добавлена ячейка T[42] = [2, 3, 1, 1, 2, 0]

[208] Добавлена ячейка T[43] = [3, 3, 1, 1, 2, 0]

[209] Добавлена ячейка T[44] = [4, 3, 1, 1, 2, 0]

[210] Добавлена ячейка T[45] = [2, 1, 2, 1, 2, 0]

[211] Добавлена ячейка T[46] = [3, 1, 2, 1, 2, 0]

[212] Добавлена ячейка T[47] = [4, 1, 2, 1, 2, 0]

[213] Добавлена ячейка T[48] = [2, 2, 2, 1, 2, 0]

[214] Добавлена ячейка T[49] = [3, 2, 2, 1, 2, 0]

[215] Добавлена ячейка T[50] = [4, 2, 2, 1, 2, 0]

[216] Добавлена ячейка T[51] = [2, 3, 2, 1, 2, 0]

[217] Добавлена ячейка T[52] = [3, 3, 2, 1, 2, 0]

[218] Добавлена ячейка T[53] = [4, 3, 2, 1, 2, 0]

[219] Добавлена ячейка T[54] = [2, 1, 0, 2, 2, 0]

[220] Добавлена ячейка T[55] = [3, 1, 0, 2, 2, 0]

[221] Добавлена ячейка T[56] = [4, 1, 0, 2, 2, 0]

[222] Добавлена ячейка T[57] = [2, 2, 0, 2, 2, 0]

[223] Добавлена ячейка T[58] = [3, 2, 0, 2, 2, 0]

[224] Добавлена ячейка T[59] = [4, 2, 0, 2, 2, 0]

[225] Добавлена ячейка T[60] = [2, 3, 0, 2, 2, 0]

[226] Добавлена ячейка T[61] = [3, 3, 0, 2, 2, 0]

[227] Добавлена ячейка T[62] = [4, 3, 0, 2, 2, 0]

[228] Добавлена ячейка T[63] = [2, 1, 1, 2, 2, 0]

[229] Добавлена ячейка T[64] = [3, 1, 1, 2, 2, 0]

[230] Добавлена ячейка T[65] = [4, 1, 1, 2, 2, 0]

[231] Добавлена ячейка T[66] = [2, 2, 1, 2, 2, 0]

[232] Добавлена ячейка T[67] = [3, 2, 1, 2, 2, 0]

[233] Добавлена ячейка T[68] = [4, 2, 1, 2, 2, 0]

[234] Добавлена ячейка T[69] = [2, 3, 1, 2, 2, 0]

[235] Добавлена ячейка T[70] = [3, 3, 1, 2, 2, 0]

[236] Добавлена ячейка T[71] = [4, 3, 1, 2, 2, 0]

[237] Добавлена ячейка T[72] = [2, 1, 2, 2, 2, 0]

[238] Добавлена ячейка T[73] = [3, 1, 2, 2, 2, 0]

[239] Добавлена ячейка T[74] = [4, 1, 2, 2, 2, 0]

[240] Добавлена ячейка T[75] = [2, 2, 2, 2, 2, 0]

[241] Добавлена ячейка T[76] = [3, 2, 2, 2, 2, 0]

[242] Добавлена ячейка T[77] = [4, 2, 2, 2, 2, 0]

[243] Добавлена ячейка T[78] = [2, 3, 2, 2, 2, 0]

[244] Добавлена ячейка T[79] = [3, 3, 2, 2, 2, 0]

[245] Добавлена ячейка T[80] = [4, 3, 2, 2, 2, 0]

[246] Добавлена ячейка T[81] = [2, 1, 0, 0, 3, 0]

[247] Добавлена ячейка T[82] = [3, 1, 0, 0, 3, 0]

[248] Добавлена ячейка T[83] = [4, 1, 0, 0, 3, 0]

[249] Добавлена ячейка T[84] = [2, 2, 0, 0, 3, 0]

[250] Добавлена ячейка T[85] = [3, 2, 0, 0, 3, 0]

[251] Добавлена ячейка T[86] = [4, 2, 0, 0, 3, 0]

[252] Добавлена ячейка T[87] = [2, 3, 0, 0, 3, 0]

[253] Добавлена ячейка T[88] = [3, 3, 0, 0, 3, 0]

[254] Добавлена ячейка T[89] = [4, 3, 0, 0, 3, 0]

[255] Добавлена ячейка T[90] = [2, 1, 1, 0, 3, 0]

[256] Добавлена ячейка T[91] = [3, 1, 1, 0, 3, 0]

[257] Добавлена ячейка T[92] = [4, 1, 1, 0, 3, 0]

[258] Добавлена ячейка T[93] = [2, 2, 1, 0, 3, 0]

[259] Добавлена ячейка T[94] = [3, 2, 1, 0, 3, 0]

[260] Добавлена ячейка T[95] = [4, 2, 1, 0, 3, 0]

[261] Добавлена ячейка T[96] = [2, 3, 1, 0, 3, 0]

[262] Добавлена ячейка T[97] = [3, 3, 1, 0, 3, 0]

[263] Добавлена ячейка T[98] = [4, 3, 1, 0, 3, 0]

[264] Добавлена ячейка T[99] = [2, 1, 2, 0, 3, 0]

[265] ....

[266] Всего множеств и ячеек: 971

[267]

[268] ----------

[269] Определение отношений доминирования по Парето и по Подиновскому

[270]

[271] [[2], [1], [1], [0], [2], [1]] доминирует [[2], [1], [0], [0], [2], [0]] по Парето

[272] [[2], [1], [1], [0], [2], [1]] доминирует [[2], [1], [1], [0], [2], [0]] по Парето

[273] [[2], [1], [1], [0], [2], [1]] доминирует [[2], [1], [0], [0], [2], [1]] по Парето

[274] [[2], [1], [2], [0], [2], [1]] доминирует [[2], [1], [1], [0], [2], [1]] по Парето

[275] [[2], [1], [2], [0], [2], [1]] доминирует [[2], [1], [0], [0], [2], [0]] по Парето

[276] [[2], [1], [2], [0], [2], [1]] доминирует [[2], [1], [1], [0], [2], [0]] по Парето

[277] [[2], [1], [2], [0], [2], [1]] доминирует [[2], [1], [2], [0], [2], [0]] по Парето

[278] [[2], [1], [2], [0], [2], [1]] доминирует [[2], [1], [0], [0], [2], [1]] по Парето

[279] [[2], [2], [1], [0], [2], [1]] доминирует [[2], [1], [1], [0], [2], [1]] по Парето

[280] [[2], [2], [1], [0], [2], [1]] доминирует [[2], [1], [0], [0], [2], [0]] по Парето

[281] [[2], [2], [1], [0], [2], [1]] доминирует [[2], [2], [0], [0], [2], [0]] по Парето

[282] [[2], [2], [1], [0], [2], [1]] доминирует [[2], [1], [1], [0], [2], [0]] по Парето

[283] [[2], [2], [1], [0], [2], [1]] доминирует [[2], [2], [1], [0], [2], [0]] по Парето

[284] [[2], [2], [1], [0], [2], [1]] доминирует [[2], [1], [0], [0], [2], [1]] по Парето

[285] [[2], [2], [1], [0], [2], [1]] доминирует [[2], [2], [0], [0], [2], [1]] по Парето

[286] [[2], [2], [1], [0], [2], [1]] доминирует [[2], [1], [1], [0], [2], [2]] по Подиновскому

[287] [[2], [3], [0, 1, 2], [1], [3], [3]] доминирует [[2], [1], [0], [0], [2], [0]] по Парето

[288] [[2], [3], [0, 1, 2], [1], [3], [3]] доминирует [[2], [2], [0], [0], [2], [0]] по Парето

[289] [[2], [3], [0, 1, 2], [1], [3], [3]] доминирует [[2], [3], [0], [0], [2], [0]] по Парето

[290] [[2], [3], [0, 1, 2], [1], [3], [3]] доминирует [[2], [1], [0], [1], [2], [0]] по Парето

[291] [[2], [3], [0, 1, 2], [1], [3], [3]] доминирует [[2], [2], [0], [1], [2], [0]] по Парето

[292] [[2], [3], [0, 1, 2], [1], [3], [3]] доминирует [[2], [3], [0], [1], [2], [0]] по Парето

[293] [[2], [3], [0, 1, 2], [1], [3], [3]] доминирует [[2], [1], [0], [0], [3], [0]] по Парето

[294] [[2], [3], [0, 1, 2], [1], [3], [3]] доминирует [[2], [2], [0], [0], [3], [0]] по Парето

[295] [[2], [3], [0, 1, 2], [1], [3], [3]] доминирует [[2], [3], [0], [0], [3], [0]] по Парето

[296] [[2], [3], [0, 1, 2], [1], [3], [3]] доминирует [[2], [1], [0], [1], [3], [0]] по Парето

[297] [[2], [3], [0, 1, 2], [1], [3], [3]] доминирует [[2], [2], [0], [1], [3], [0]] по Парето

[298] [[2], [3], [0, 1, 2], [1], [3], [3]] доминирует [[2], [3], [0], [1], [3], [0]] по Парето

[299] [[2], [3], [0, 1, 2], [1], [3], [3]] доминирует [[2], [1], [0], [0], [2], [1]] по Парето

[300] [[2], [3], [0, 1, 2], [1], [3], [3]] доминирует [[2], [2], [0], [0], [2], [1]] по Парето

[301] [[2], [3], [0, 1, 2], [1], [3], [3]] доминирует [[2], [3], [0], [0], [2], [1]] по Парето

[302] [[2], [3], [0, 1, 2], [1], [3], [3]] доминирует [[2], [1], [0], [1], [2], [1]] по Парето

[303] [[2], [3], [0, 1, 2], [1], [3], [3]] доминирует [[2], [2], [0], [1], [2], [1]] по Парето

[304] [[2], [3], [0, 1, 2], [1], [3], [3]] доминирует [[2], [3], [0], [1], [2], [1]] по Парето

[305] [[2], [3], [0, 1, 2], [1], [3], [3]] доминирует [[2], [1], [0], [0], [3], [1]] по Парето

[306] [[2], [3], [0, 1, 2], [1], [3], [3]] доминирует [[2], [2], [0], [0], [3], [1]] по Парето

[307] [[2], [3], [0, 1, 2], [1], [3], [3]] доминирует [[2], [3], [0], [0], [3], [1]] по Парето

[308] [[2], [3], [0, 1, 2], [1], [3], [3]] доминирует [[2], [1], [0], [1], [3], [1]] по Парето

[309] [[2], [3], [0, 1, 2], [1], [3], [3]] доминирует [[2], [2], [0], [1], [3], [1]] по Парето

[310] [[2], [3], [0, 1, 2], [1], [3], [3]] доминирует [[2], [3], [0], [1], [3], [1]] по Парето

[311] [[2], [3], [0, 1, 2], [1], [3], [3]] доминирует [[2], [1], [0], [0], [2], [2]] по Парето

[312] [[2], [3], [0, 1, 2], [1], [3], [3]] доминирует [[2], [2], [0], [0], [2], [2]] по Парето

[313] [[2], [3], [0, 1, 2], [1], [3], [3]] доминирует [[2], [3], [0], [0], [2], [2]] по Парето

[314] [[2], [3], [0, 1, 2], [1], [3], [3]] доминирует [[2], [1], [0], [1], [2], [2]] по Парето

[315] [[2], [3], [0, 1, 2], [1], [3], [3]] доминирует [[2], [2], [0], [1], [2], [2]] по Парето

[316] [[2], [3], [0, 1, 2], [1], [3], [3]] доминирует [[2], [3], [0], [1], [2], [2]] по Парето

[317] [[2], [3], [0, 1, 2], [1], [3], [3]] доминирует [[2], [1], [0], [0], [3], [2]] по Парето

[318] [[2], [3], [0, 1, 2], [1], [3], [3]] доминирует [[2], [2], [0], [0], [3], [2]] по Парето

[319] [[2], [3], [0, 1, 2], [1], [3], [3]] доминирует [[2], [3], [0], [0], [3], [2]] по Парето

[320] [[2], [3], [0, 1, 2], [1], [3], [3]] доминирует [[2], [1], [0], [1], [3], [2]] по Парето

[321] [[2], [3], [0, 1, 2], [1], [3], [3]] доминирует [[2], [2], [0], [1], [3], [2]] по Парето

[322] [[2], [3], [0, 1, 2], [1], [3], [3]] доминирует [[2], [3], [0], [1], [3], [2]] по Парето

[323] [[2], [3], [0, 1, 2], [1], [3], [3]] доминирует [[2], [1], [0], [0], [2], [3]] по Парето

[324] [[2], [3], [0, 1, 2], [1], [3], [3]] доминирует [[2], [2], [0], [0], [2], [3]] по Парето

[325] [[2], [3], [0, 1, 2], [1], [3], [3]] доминирует [[2], [3], [0], [0], [2], [3]] по Парето

[326] [[2], [3], [0, 1, 2], [1], [3], [3]] доминирует [[2], [1], [0], [1], [2], [3]] по Парето

[327] [[2], [3], [0, 1, 2], [1], [3], [3]] доминирует [[2], [2], [0], [1], [2], [3]] по Парето

[328] [[2], [3], [0, 1, 2], [1], [3], [3]] доминирует [[2], [3], [0], [1], [2], [3]] по Парето

[329] [[2], [3], [0, 1, 2], [1], [3], [3]] доминирует [[2], [1], [0], [0], [3], [3]] по Парето

[330] [[2], [3], [0, 1, 2], [1], [3], [3]] доминирует [[2], [2], [0], [0], [3], [3]] по Парето

[331] [[2], [3], [0, 1, 2], [1], [3], [3]] доминирует [[2], [3], [0], [0], [3], [3]] по Парето

[332] [[2], [3], [0, 1, 2], [1], [3], [3]] доминирует [[2], [3], [1], [0], [3], [3]] по Подиновскому

[333] [[2], [3], [0, 1, 2], [1], [3], [3]] доминирует [[2], [1], [0], [1], [3], [3]] по Парето

[334] [[2], [3], [0, 1, 2], [1], [3], [3]] доминирует [[2], [2], [0], [1], [3], [3]] по Парето

[335] [[4], [2], [0, 1, 2], [0], [4], [2]] доминирует [[2], [1], [0], [0], [2], [0]] по Парето

[336] [[4], [2], [0, 1, 2], [0], [4], [2]] доминирует [[3], [1], [0], [0], [2], [0]] по Парето

[337] [[4], [2], [0, 1, 2], [0], [4], [2]] доминирует [[4], [1], [0], [0], [2], [0]] по Парето

[338] [[4], [2], [0, 1, 2], [0], [4], [2]] доминирует [[2], [2], [0], [0], [2], [0]] по Парето

[339] [[4], [2], [0, 1, 2], [0], [4], [2]] доминирует [[3], [2], [0], [0], [2], [0]] по Парето

[340] [[4], [2], [0, 1, 2], [0], [4], [2]] доминирует [[4], [2], [0], [0], [2], [0]] по Парето

[341] [[4], [2], [0, 1, 2], [0], [4], [2]] доминирует [[2], [1], [0], [0], [3], [0]] по Парето

[342] [[4], [2], [0, 1, 2], [0], [4], [2]] доминирует [[3], [1], [0], [0], [3], [0]] по Парето

[343] [[4], [2], [0, 1, 2], [0], [4], [2]] доминирует [[4], [1], [0], [0], [3], [0]] по Парето

[344] [[4], [2], [0, 1, 2], [0], [4], [2]] доминирует [[2], [2], [0], [0], [3], [0]] по Парето

[345] [[4], [2], [0, 1, 2], [0], [4], [2]] доминирует [[3], [2], [0], [0], [3], [0]] по Парето

[346] [[4], [2], [0, 1, 2], [0], [4], [2]] доминирует [[4], [2], [0], [0], [3], [0]] по Парето

[347] [[4], [2], [0, 1, 2], [0], [4], [2]] доминирует [[2], [1], [0], [0], [4], [0]] по Парето

[348] [[4], [2], [0, 1, 2], [0], [4], [2]] доминирует [[3], [1], [0], [0], [4], [0]] по Парето

[349] [[4], [2], [0, 1, 2], [0], [4], [2]] доминирует [[4], [1], [0], [0], [4], [0]] по Парето

[350] [[4], [2], [0, 1, 2], [0], [4], [2]] доминирует [[2], [2], [0], [0], [4], [0]] по Парето

[351] [[4], [2], [0, 1, 2], [0], [4], [2]] доминирует [[3], [2], [0], [0], [4], [0]] по Парето

[352] [[4], [2], [0, 1, 2], [0], [4], [2]] доминирует [[4], [2], [0], [0], [4], [0]] по Парето

[353] [[4], [2], [0, 1, 2], [0], [4], [2]] доминирует [[2], [1], [0], [0], [2], [1]] по Парето

[354] [[4], [2], [0, 1, 2], [0], [4], [2]] доминирует [[3], [1], [0], [0], [2], [1]] по Парето

[355] [[4], [2], [0, 1, 2], [0], [4], [2]] доминирует [[4], [1], [0], [0], [2], [1]] по Парето

[356] [[4], [2], [0, 1, 2], [0], [4], [2]] доминирует [[2], [2], [0], [0], [2], [1]] по Парето

[357] [[4], [2], [0, 1, 2], [0], [4], [2]] доминирует [[3], [2], [0], [0], [2], [1]] по Парето

[358] [[4], [2], [0, 1, 2], [0], [4], [2]] доминирует [[4], [2], [0], [0], [2], [1]] по Парето

[359] [[4], [2], [0, 1, 2], [0], [4], [2]] доминирует [[2], [1], [0], [0], [3], [1]] по Парето

[360] [[4], [2], [0, 1, 2], [0], [4], [2]] доминирует [[3], [1], [0], [0], [3], [1]] по Парето

[361] [[4], [2], [0, 1, 2], [0], [4], [2]] доминирует [[4], [1], [0], [0], [3], [1]] по Парето

[362] [[4], [2], [0, 1, 2], [0], [4], [2]] доминирует [[2], [2], [0], [0], [3], [1]] по Парето

[363] [[4], [2], [0, 1, 2], [0], [4], [2]] доминирует [[3], [2], [0], [0], [3], [1]] по Парето

[364] [[4], [2], [0, 1, 2], [0], [4], [2]] доминирует [[4], [2], [0], [0], [3], [1]] по Парето

[365] [[4], [2], [0, 1, 2], [0], [4], [2]] доминирует [[2], [1], [0], [0], [4], [1]] по Парето

[366] [[4], [2], [0, 1, 2], [0], [4], [2]] доминирует [[3], [1], [0], [0], [4], [1]] по Парето

[367] [[4], [2], [0, 1, 2], [0], [4], [2]] доминирует [[4], [1], [0], [0], [4], [1]] по Парето

[368] [[4], [2], [0, 1, 2], [0], [4], [2]] доминирует [[2], [2], [0], [0], [4], [1]] по Парето

[369] [[4], [2], [0, 1, 2], [0], [4], [2]] доминирует [[3], [2], [0], [0], [4], [1]] по Парето

[370] [[4], [2], [0, 1, 2], [0], [4], [2]] доминирует [[4], [2], [0], [0], [4], [1]] по Парето

[371] [[4], [2], [0, 1, 2], [0], [4], [2]] доминирует [[2], [1], [0], [0], [2], [2]] по Парето

[372] [[2], [1], [0], [1], [2], [0]] доминирует [[2], [1], [1], [0], [2], [0]] по Подиновскому

[373] [[3], [1], [0], [1], [2], [0]] доминирует [[3], [1], [1], [0], [2], [0]] по Подиновскому

[374] [[4], [1], [0], [1], [2], [0]] доминирует [[4], [1], [1], [0], [2], [0]] по Подиновскому

[375] [[2], [2], [0], [1], [2], [0]] доминирует [[2], [2], [1], [0], [2], [0]] по Подиновскому

[376] [[3], [2], [0], [1], [2], [0]] доминирует [[3], [2], [1], [0], [2], [0]] по Подиновскому

[377] [[4], [2], [0], [1], [2], [0]] доминирует [[4], [2], [1], [0], [2], [0]] по Подиновскому

[378] [[2], [3], [0], [1], [2], [0]] доминирует [[2], [3], [1], [0], [2], [0]] по Подиновскому

[379] [[3], [3], [0], [1], [2], [0]] доминирует [[3], [3], [1], [0], [2], [0]] по Подиновскому

[380] [[4], [3], [0], [1], [2], [0]] доминирует [[4], [3], [1], [0], [2], [0]] по Подиновскому

[381] [[2], [1], [0], [2], [2], [0]] доминирует [[2], [1], [2], [0], [2], [0]] по Подиновскому

[382] [[3], [1], [0], [2], [2], [0]] доминирует [[3], [1], [2], [0], [2], [0]] по Подиновскому

[383] [[4], [1], [0], [2], [2], [0]] доминирует [[4], [1], [2], [0], [2], [0]] по Подиновскому

[384] [[2], [2], [0], [2], [2], [0]] доминирует [[2], [2], [2], [0], [2], [0]] по Подиновскому

[385] [[3], [2], [0], [2], [2], [0]] доминирует [[3], [2], [2], [0], [2], [0]] по Подиновскому

[386] [[4], [2], [0], [2], [2], [0]] доминирует [[4], [2], [2], [0], [2], [0]] по Подиновскому

[387] [[2], [3], [0], [2], [2], [0]] доминирует [[2], [3], [2], [0], [2], [0]] по Подиновскому

[388] [[3], [3], [0], [2], [2], [0]] доминирует [[3], [3], [2], [0], [2], [0]] по Подиновскому

[389] [[4], [3], [0], [2], [2], [0]] доминирует [[4], [3], [2], [0], [2], [0]] по Подиновскому

[390] [[2], [1], [1], [2], [2], [0]] доминирует [[2], [1], [2], [1], [2], [0]] по Подиновскому

[391] [[3], [1], [1], [2], [2], [0]] доминирует [[3], [1], [2], [1], [2], [0]] по Подиновскому

[392] [[4], [1], [1], [2], [2], [0]] доминирует [[4], [1], [2], [1], [2], [0]] по Подиновскому

[393] [[2], [2], [1], [2], [2], [0]] доминирует [[2], [2], [2], [1], [2], [0]] по Подиновскому

[394] [[3], [2], [1], [2], [2], [0]] доминирует [[3], [2], [2], [1], [2], [0]] по Подиновскому

[395] [[4], [2], [1], [2], [2], [0]] доминирует [[4], [2], [2], [1], [2], [0]] по Подиновскому

[396] [[2], [3], [1], [2], [2], [0]] доминирует [[2], [3], [2], [1], [2], [0]] по Подиновскому

[397] [[3], [3], [1], [2], [2], [0]] доминирует [[3], [3], [2], [1], [2], [0]] по Подиновскому

[398] [[4], [3], [1], [2], [2], [0]] доминирует [[4], [3], [2], [1], [2], [0]] по Подиновскому

[399] [[2], [1], [0], [1], [3], [0]] доминирует [[2], [1], [1], [0], [3], [0]] по Подиновскому

[400] [[3], [1], [0], [1], [3], [0]] доминирует [[3], [1], [1], [0], [3], [0]] по Подиновскому

[401] [[4], [1], [0], [1], [3], [0]] доминирует [[4], [1], [1], [0], [3], [0]] по Подиновскому

[402] [[2], [2], [0], [1], [3], [0]] доминирует [[2], [2], [1], [0], [3], [0]] по Подиновскому

[403] [[3], [2], [0], [1], [3], [0]] доминирует [[3], [2], [1], [0], [3], [0]] по Подиновскому

[404] [[4], [2], [0], [1], [3], [0]] доминирует [[4], [2], [1], [0], [3], [0]] по Подиновскому

[405] [[2], [3], [0], [1], [3], [0]] доминирует [[2], [3], [1], [0], [3], [0]] по Подиновскому

[406] [[3], [3], [0], [1], [3], [0]] доминирует [[3], [3], [1], [0], [3], [0]] по Подиновскому

[407] [[4], [3], [0], [1], [3], [0]] доминирует [[4], [3], [1], [0], [3], [0]] по Подиновскому

[408] [[2], [1], [0], [2], [3], [0]] доминирует [[2], [1], [2], [0], [3], [0]] по Подиновскому

[409] [[3], [1], [0], [2], [3], [0]] доминирует [[3], [1], [2], [0], [3], [0]] по Подиновскому

[410] [[4], [1], [0], [2], [3], [0]] доминирует [[4], [1], [2], [0], [3], [0]] по Подиновскому

[411] [[2], [2], [0], [2], [3], [0]] доминирует [[2], [2], [2], [0], [3], [0]] по Подиновскому

[412] [[3], [2], [0], [2], [3], [0]] доминирует [[3], [2], [2], [0], [3], [0]] по Подиновскому

[413] [[4], [2], [0], [2], [3], [0]] доминирует [[4], [2], [2], [0], [3], [0]] по Подиновскому

[414] [[2], [3], [0], [2], [3], [0]] доминирует [[2], [3], [2], [0], [3], [0]] по Подиновскому

[415] [[3], [3], [0], [2], [3], [0]] доминирует [[3], [3], [2], [0], [3], [0]] по Подиновскому

[416] [[4], [3], [0], [2], [3], [0]] доминирует [[4], [3], [2], [0], [3], [0]] по Подиновскому

[417] [[2], [1], [1], [2], [3], [0]] доминирует [[2], [1], [2], [1], [3], [0]] по Подиновскому

[418] [[3], [1], [1], [2], [3], [0]] доминирует [[3], [1], [2], [1], [3], [0]] по Подиновскому

[419] [[4], [1], [1], [2], [3], [0]] доминирует [[4], [1], [2], [1], [3], [0]] по Подиновскому

[420] [[2], [2], [1], [2], [3], [0]] доминирует [[2], [2], [2], [1], [3], [0]] по Подиновскому

[421] [[3], [2], [1], [2], [3], [0]] доминирует [[3], [2], [2], [1], [3], [0]] по Подиновскому

[422] [[4], [2], [1], [2], [3], [0]] доминирует [[4], [2], [2], [1], [3], [0]] по Подиновскому

[423] [[2], [3], [1], [2], [3], [0]] доминирует [[2], [3], [2], [1], [3], [0]] по Подиновскому

[424] [[3], [3], [1], [2], [3], [0]] доминирует [[3], [3], [2], [1], [3], [0]] по Подиновскому

[425] [[4], [3], [1], [2], [3], [0]] доминирует [[4], [3], [2], [1], [3], [0]] по Подиновскому

[426] [[2], [1], [0], [1], [4], [0]] доминирует [[2], [1], [1], [0], [4], [0]] по Подиновскому

[427] [[3], [1], [0], [1], [4], [0]] доминирует [[3], [1], [1], [0], [4], [0]] по Подиновскому

[428] [[4], [1], [0], [1], [4], [0]] доминирует [[4], [1], [1], [0], [4], [0]] по Подиновскому

[429] [[2], [2], [0], [1], [4], [0]] доминирует [[2], [2], [1], [0], [4], [0]] по Подиновскому

[430] [[3], [2], [0], [1], [4], [0]] доминирует [[3], [2], [1], [0], [4], [0]] по Подиновскому

[431] [[4], [2], [0], [1], [4], [0]] доминирует [[4], [2], [1], [0], [4], [0]] по Подиновскому

[432] [[2], [3], [0], [1], [4], [0]] доминирует [[2], [3], [1], [0], [4], [0]] по Подиновскому

[433] [[3], [3], [0], [1], [4], [0]] доминирует [[3], [3], [1], [0], [4], [0]] по Подиновскому

[434] [[4], [3], [0], [1], [4], [0]] доминирует [[4], [3], [1], [0], [4], [0]] по Подиновскому

[435] [[2], [1], [0], [2], [4], [0]] доминирует [[2], [1], [2], [0], [4], [0]] по Подиновскому

[436] [[3], [1], [0], [2], [4], [0]] доминирует [[3], [1], [2], [0], [4], [0]] по Подиновскому

[437] [[4], [1], [0], [2], [4], [0]] доминирует [[4], [1], [2], [0], [4], [0]] по Подиновскому

[438] [[2], [2], [0], [2], [4], [0]] доминирует [[2], [2], [2], [0], [4], [0]] по Подиновскому

[439] [[3], [2], [0], [2], [4], [0]] доминирует [[3], [2], [2], [0], [4], [0]] по Подиновскому

[440] [[4], [2], [0], [2], [4], [0]] доминирует [[4], [2], [2], [0], [4], [0]] по Подиновскому

[441] [[2], [3], [0], [2], [4], [0]] доминирует [[2], [3], [2], [0], [4], [0]] по Подиновскому

[442] [[3], [3], [0], [2], [4], [0]] доминирует [[3], [3], [2], [0], [4], [0]] по Подиновскому

[443] [[4], [3], [0], [2], [4], [0]] доминирует [[4], [3], [2], [0], [4], [0]] по Подиновскому

[444] [[2], [1], [1], [2], [4], [0]] доминирует [[2], [1], [2], [1], [4], [0]] по Подиновскому

[445] [[3], [1], [1], [2], [4], [0]] доминирует [[3], [1], [2], [1], [4], [0]] по Подиновскому

[446] [[4], [1], [1], [2], [4], [0]] доминирует [[4], [1], [2], [1], [4], [0]] по Подиновскому

[447] [[2], [2], [1], [2], [4], [0]] доминирует [[2], [2], [2], [1], [4], [0]] по Подиновскому

[448] [[3], [2], [1], [2], [4], [0]] доминирует [[3], [2], [2], [1], [4], [0]] по Подиновскому

[449] [[4], [2], [1], [2], [4], [0]] доминирует [[4], [2], [2], [1], [4], [0]] по Подиновскому

[450] [[2], [3], [1], [2], [4], [0]] доминирует [[2], [3], [2], [1], [4], [0]] по Подиновскому

[451] [[3], [3], [1], [2], [4], [0]] доминирует [[3], [3], [2], [1], [4], [0]] по Подиновскому

[452] [[4], [3], [1], [2], [4], [0]] доминирует [[4], [3], [2], [1], [4], [0]] по Подиновскому

[453] [[4], [3], [2], [2], [4], [0]] доминирует [[0, 1, 2, 3, 4], [0], [0, 1, 2], [0, 1, 2], [0, 1, 2, 3, 4], [0, 1, 2, 3]] по Подиновскому

[454] [[2], [2], [0], [0], [2], [1]] доминирует [[2], [1], [0], [0], [2], [2]] по Подиновскому

[455] [[3], [2], [0], [0], [2], [1]] доминирует [[3], [1], [0], [0], [2], [2]] по Подиновскому

[456] [[4], [2], [0], [0], [2], [1]] доминирует [[4], [1], [0], [0], [2], [2]] по Подиновскому

[457] [[2], [3], [0], [0], [2], [1]] доминирует [[2], [1], [0], [0], [2], [3]] по Подиновскому

[458] [[3], [3], [0], [0], [2], [1]] доминирует [[3], [1], [0], [0], [2], [3]] по Подиновскому

[459] [[4], [3], [0], [0], [2], [1]] доминирует [[4], [1], [0], [0], [2], [3]] по Подиновскому

[460] [[3], [2], [1], [0], [2], [1]] доминирует [[3], [1], [1], [0], [2], [2]] по Подиновскому

[461] [[4], [2], [1], [0], [2], [1]] доминирует [[4], [1], [1], [0], [2], [2]] по Подиновскому

[462] [[2], [3], [1], [0], [2], [1]] доминирует [[2], [1], [1], [0], [2], [3]] по Подиновскому

[463] [[3], [3], [1], [0], [2], [1]] доминирует [[3], [1], [1], [0], [2], [3]] по Подиновскому

[464] [[4], [3], [1], [0], [2], [1]] доминирует [[4], [1], [1], [0], [2], [3]] по Подиновскому

[465] [[2], [2], [2], [0], [2], [1]] доминирует [[2], [1], [2], [0], [2], [2]] по Подиновскому

[466] [[3], [2], [2], [0], [2], [1]] доминирует [[3], [1], [2], [0], [2], [2]] по Подиновскому

[467] [[4], [2], [2], [0], [2], [1]] доминирует [[4], [1], [2], [0], [2], [2]] по Подиновскому

[468] [[2], [3], [2], [0], [2], [1]] доминирует [[2], [1], [2], [0], [2], [3]] по Подиновскому

[469] .....

[470] Отношений доминирования по Парето: 76181, по Подиновскому 567

[471] Дополним граф отношениями доминирования, введенными пользователем

[472] [[2], [1], [1], [0], [2], [1]] доминирует [[0, 1], [0, 1, 2, 3], [0, 1, 2], [0, 1, 2], [0, 1, 2, 3, 4], [0,

1, 2, 3]]

[473] [[2], [1], [1], [0], [2], [1]] доминирует [[0, 1, 2, 3, 4], [0], [0, 1, 2], [0, 1, 2], [0, 1, 2, 3, 4], [0, 1, 2, 3]]

[474] [[2], [1], [1], [0], [2], [1]] доминирует [[0, 1, 2, 3, 4], [0, 1, 2, 3], [0, 1, 2], [0, 1, 2], [0, 1], [0, 1, 2, 3]]

[475] [[2], [1], [2], [0], [2], [1]] доминирует [[0, 1], [0, 1, 2, 3], [0, 1, 2], [0, 1, 2], [0, 1, 2, 3, 4], [0, 1, 2, 3]]

[476] [[2], [1], [2], [0], [2], [1]] доминирует [[0, 1, 2, 3, 4], [0], [0, 1, 2], [0, 1, 2], [0, 1, 2, 3, 4], [0, 1, 2, 3]]

[477] [[2], [1], [2], [0], [2], [1]] доминирует [[0, 1, 2, 3, 4], [0, 1, 2, 3], [0, 1, 2], [0, 1, 2], [0, 1], [0, 1, 2, 3]]

[478] [[2], [1], [2], [0], [2], [1]] доминирует [[2], [1], [1], [0], [2], [1]]

[479] [[2], [2], [1], [0], [2], [1]] доминирует [[0, 1], [0, 1, 2, 3], [0, 1, 2], [0, 1, 2], [0, 1, 2, 3, 4], [0, 1, 2, 3]]

[480] [[2], [2], [1], [0], [2], [1]] доминирует [[0, 1, 2, 3, 4], [0], [0, 1, 2], [0, 1, 2], [0, 1, 2, 3, 4], [0, 1, 2, 3]]

[481] [[2], [2], [1], [0], [2], [1]] доминирует [[0, 1, 2, 3, 4], [0, 1, 2, 3], [0, 1, 2], [0, 1, 2], [0, 1], [0, 1, 2, 3]]

[482] [[2], [2], [1], [0], [2], [1]] доминирует [[2], [1], [1], [0], [2], [1]]

[483] [[2], [3], [0, 1, 2], [1], [3], [3]] доминирует [[0, 1], [0, 1, 2, 3], [0, 1, 2], [0, 1, 2], [0, 1, 2, 3, 4], [0, 1, 2, 3]]

[484] [[2], [3], [0, 1, 2], [1], [3], [3]] доминирует [[0, 1, 2, 3, 4], [0], [0, 1, 2], [0, 1, 2], [0, 1, 2, 3, 4], [0, 1, 2, 3]]

[485] [[2], [3], [0, 1, 2], [1], [3], [3]] доминирует [[0, 1, 2, 3, 4], [0, 1, 2, 3], [0, 1, 2], [0, 1, 2], [0, 1], [0, 1, 2, 3]]

[486] [[2], [3], [0, 1, 2], [1], [3], [3]] доминирует [[2], [1], [1], [0], [2], [1]]

[487] [[2], [3], [0, 1, 2], [1], [3], [3]] доминирует [[2], [1], [2], [0], [2], [1]]

[488] [[2], [3], [0, 1, 2], [1], [3], [3]] доминирует [[2], [2], [1], [0], [2], [1]]

[489] [[4], [2], [0, 1, 2], [0], [4], [2]] доминирует [[0, 1], [0, 1, 2, 3], [0, 1, 2], [0, 1, 2], [0, 1, 2, 3, 4], [0, 1, 2, 3]]

[490] [[4], [2], [0, 1, 2], [0], [4], [2]] доминирует [[0, 1, 2, 3, 4], [0], [0, 1, 2], [0, 1, 2], [0, 1, 2, 3, 4], [0, 1, 2, 3]]

[491] [[4], [2], [0, 1, 2], [0], [4], [2]] доминирует [[0, 1, 2, 3, 4], [0, 1, 2, 3], [0, 1, 2], [0, 1, 2], [0, 1], [0, 1, 2, 3]]

[492] [[4], [2], [0, 1, 2], [0], [4], [2]] доминирует [[2], [1], [1], [0], [2], [1]]

[493] [[4], [2], [0, 1, 2], [0], [4], [2]] доминирует [[2], [1], [2], [0], [2], [1]]

[494] [[4], [2], [0, 1, 2], [0], [4], [2]] доминирует [[2], [2], [1], [0], [2], [1]]

[495] Отношений доминирования по Парето: 76181, по Подиновскому 567, пользовательских 23

[496]

[497] 970 Недоминируемая вершина: [[4], [3], [2], [2], [4], [3]]

[498] Всего дуг в графе доминирования: 76769

[499]

[500] ----------

[501] Поиск циклов

[502] Анализируется путь [0, 968, 969, 970]

[503] Анализируется путь [0, 968, 970]

[504] Анализируется путь [0, 969]

[505] Анализируется путь [0, 970]

[506] Анализируется путь [0, 3, 4, 267, 268, 271, 274, 301, 319, 328, 409, 490, 730, 970]

[507] Анализируется путь [0, 3, 4, 267, 268, 271, 274, 301, 319, 328, 409, 490, 970]

[508] Анализируется путь [0, 3, 4, 267, 268, 271, 274, 301, 319, 328, 409, 652, 730]

[509] Анализируется путь [0, 3, 4, 267, 268, 271, 274, 301, 319, 328, 409, 652, 889, 970]

[510] Анализируется путь [0, 3, 4, 267, 268, 271, 274, 301, 319, 328, 409, 652, 970]

[511] Анализируется путь [0, 3, 4, 267, 268, 271, 274, 301, 319, 328, 409, 730]

[512] Анализируется путь [0, 3, 4, 267, 268, 271, 274, 301, 319, 328, 409, 889]

[513] Анализируется путь [0, 3, 4, 267, 268, 271, 274, 301, 319, 328, 409, 970]

[514] Анализируется путь [0, 3, 4, 267, 268, 271, 274, 301, 319, 328, 490]

[515] Анализируется путь [0, 3, 4, 267, 268, 271, 274, 301, 319, 328, 571, 652]

[516] Анализируется путь [0, 3, 4, 267, 268, 271, 274, 301, 319, 328, 571, 730]

[517] Анализируется путь [0, 3, 4, 267, 268, 271, 274, 301, 319, 328, 571, 811, 889]

[518] Анализируется путь [0, 3, 4, 267, 268, 271, 274, 301, 319, 328, 571, 811, 970]

[519] Анализируется путь [0, 3, 4, 267, 268, 271, 274, 301, 319, 328, 571, 889]

[520] Анализируется путь [0, 3, 4, 267, 268, 271, 274, 301, 319, 328, 571, 970]

[521] Анализируется путь [0, 3, 4, 267, 268, 271, 274, 301, 319, 328, 652]

[522] Анализируется путь [0, 3, 4, 267, 268, 271, 274, 301, 319, 328, 730]

[523] Анализируется путь [0, 3, 4, 267, 268, 271, 274, 301, 319, 328, 811]

[524] Анализируется путь [0, 3, 4, 267, 268, 271, 274, 301, 319, 328, 889]

[525] Анализируется путь [0, 3, 4, 267, 268, 271, 274, 301, 319, 328, 970]

[526] Анализируется путь [0, 3, 4, 267, 268, 271, 274, 301, 319, 400, 409]

[527] Анализируется путь [0, 3, 4, 267, 268, 271, 274, 301, 319, 400, 481, 490]

[528] Анализируется путь [0, 3, 4, 267, 268, 271, 274, 301, 319, 400, 481, 721, 730]

[529] Анализируется путь [0, 3, 4, 267, 268, 271, 274, 301, 319, 400, 481, 721, 961, 970]

[530] Анализируется путь [0, 3, 4, 267, 268, 271, 274, 301, 319, 400, 481, 721, 970]

[531] Анализируется путь [0, 3, 4, 267, 268, 271, 274, 301, 319, 400, 481, 730]

[532] Анализируется путь [0, 3, 4, 267, 268, 271, 274, 301, 319, 400, 481, 961]

[533] Анализируется путь [0, 3, 4, 267, 268, 271, 274, 301, 319, 400, 481, 970]

[534] Анализируется путь [0, 3, 4, 267, 268, 271, 274, 301, 319, 400, 490]

[535] Анализируется путь [0, 3, 4, 267, 268, 271, 274, 301, 319, 400, 643, 652]

[536] Анализируется путь [0, 3, 4, 267, 268, 271, 274, 301, 319, 400, 643, 721]

[537] Анализируется путь [0, 3, 4, 267, 268, 271, 274, 301, 319, 400, 643, 730]

[538] Анализируется путь [0, 3, 4, 267, 268, 271, 274, 301, 319, 400, 643, 880, 889]

[539] Анализируется путь [0, 3, 4, 267, 268, 271, 274, 301, 319, 400, 643, 880, 961]

[540] Анализируется путь [0, 3, 4, 267, 268, 271, 274, 301, 319, 400, 643, 880, 970]

[541] Анализируется путь [0, 3, 4, 267, 268, 271, 274, 301, 319, 400, 643, 889]

[542] Анализируется путь [0, 3, 4, 267, 268, 271, 274, 301, 319, 400, 643, 961]

[543] Анализируется путь [0, 3, 4, 267, 268, 271, 274, 301, 319, 400, 643, 970]

[544] Анализируется путь [0, 3, 4, 267, 268, 271, 274, 301, 319, 400, 652]

[545] Анализируется путь [0, 3, 4, 267, 268, 271, 274, 301, 319, 400, 721]

[546] Анализируется путь [0, 3, 4, 267, 268, 271, 274, 301, 319, 400, 730]

[547] Анализируется путь [0, 3, 4, 267, 268, 271, 274, 301, 319, 400, 880]

[548] Анализируется путь [0, 3, 4, 267, 268, 271, 274, 301, 319, 400, 889]

[549] Анализируется путь [0, 3, 4, 267, 268, 271, 274, 301, 319, 400, 961]

[550] Анализируется путь [0, 3, 4, 267, 268, 271, 274, 301, 319, 400, 970]

[551] Анализируется путь [0, 3, 4, 267, 268, 271, 274, 301, 319, 409]

[552] Анализируется путь [0, 3, 4, 267, 268, 271, 274, 301, 319, 481]

[553] Анализируется путь [0, 3, 4, 267, 268, 271, 274, 301, 319, 490]

[554] Анализируется путь [0, 3, 4, 267, 268, 271, 274, 301, 319, 562, 571]

[555] Анализируется путь [0, 3, 4, 267, 268, 271, 274, 301, 319, 562, 643]

[556] Анализируется путь [0, 3, 4, 267, 268, 271, 274, 301, 319, 562, 652]

[557] Анализируется путь [0, 3, 4, 267, 268, 271, 274, 301, 319, 562, 721]

[558] Анализируется путь [0, 3, 4, 267, 268, 271, 274, 301, 319, 562, 730]

[559] Анализируется путь [0, 3, 4, 267, 268, 271, 274, 301, 319, 562, 802, 811]

[560] Анализируется путь [0, 3, 4, 267, 268, 271, 274, 301, 319, 562, 802, 880]

[561] Анализируется путь [0, 3, 4, 267, 268, 271, 274, 301, 319, 562, 802, 889]

[562] Анализируется путь [0, 3, 4, 267, 268, 271, 274, 301, 319, 562, 802, 961]

[563] Анализируется путь [0, 3, 4, 267, 268, 271, 274, 301, 319, 562, 802, 970]

[564] Анализируется путь [0, 3, 4, 267, 268, 271, 274, 301, 319, 562, 811]

[565] Анализируется путь [0, 3, 4, 267, 268, 271, 274, 301, 319, 562, 880]

[566] Анализируется путь [0, 3, 4, 267, 268, 271, 274, 301, 319, 562, 889]

[567] Анализируется путь [0, 3, 4, 267, 268, 271, 274, 301, 319, 562, 961]

[568] Анализируется путь [0, 3, 4, 267, 268, 271, 274, 301, 319, 562, 970]

[569] Анализируется путь [0, 3, 4, 267, 268, 271, 274, 301, 319, 571]

[570] Анализируется путь [0, 3, 4, 267, 268, 271, 274, 301, 319, 643]

[571] Анализируется путь [0, 3, 4, 267, 268, 271, 274, 301, 319, 652]

[572] Анализируется путь [0, 3, 4, 267, 268, 271, 274, 301, 319, 721]

[573] Анализируется путь [0, 3, 4, 267, 268, 271, 274, 301, 319, 730]

[574] Анализируется путь [0, 3, 4, 267, 268, 271, 274, 301, 319, 802]

[575] Анализируется путь [0, 3, 4, 267, 268, 271, 274, 301, 319, 811]

[576] Анализируется путь [0, 3, 4, 267, 268, 271, 274, 301, 319, 880]

[577] Анализируется путь [0, 3, 4, 267, 268, 271, 274, 301, 319, 889]

[578] Анализируется путь [0, 3, 4, 267, 268, 271, 274, 301, 319, 961]

[579] Анализируется путь [0, 3, 4, 267, 268, 271, 274, 301, 319, 970]

[580] Анализируется путь [0, 3, 4, 267, 268, 271, 274, 301, 328]

[581] Анализируется путь [0, 3, 4, 267, 268, 271, 274, 301, 382, 400]

[582] Анализируется путь [0, 3, 4, 267, 268, 271, 274, 301, 382, 409]

[583] Анализируется путь [0, 3, 4, 267, 268, 271, 274, 301, 382, 463, 481]

[584] Анализируется путь [0, 3, 4, 267, 268, 271, 274, 301, 382, 463, 490]

[585] Анализируется путь [0, 3, 4, 267, 268, 271, 274, 301, 382, 463, 703, 721]

[586] Анализируется путь [0, 3, 4, 267, 268, 271, 274, 301, 382, 463, 703, 730]

[587] Анализируется путь [0, 3, 4, 267, 268, 271, 274, 301, 382, 463, 703, 943, 961]

[588] Анализируется путь [0, 3, 4, 267, 268, 271, 274, 301, 382, 463, 703, 943, 970]

[589] Анализируется путь [0, 3, 4, 267, 268, 271, 274, 301, 382, 463, 703, 970]

[590] Анализируется путь [0, 3, 4, 267, 268, 271, 274, 301, 382, 463, 730]

[591] Анализируется путь [0, 3, 4, 267, 268, 271, 274, 301, 382, 463, 943]

[592] Анализируется путь [0, 3, 4, 267, 268, 271, 274, 301, 382, 463, 970]

[593] Анализируется путь [0, 3, 4, 267, 268, 271, 274, 301, 382, 490]

[594] Анализируется путь [0, 3, 4, 267, 268, 271, 274, 301, 382, 625, 643]

[595] Анализируется путь [0, 3, 4, 267, 268, 271, 274, 301, 382, 625, 652]

[596] Анализируется путь [0, 3, 4, 267, 268, 271, 274, 301, 382, 625, 703]

[597] Анализируется путь [0, 3, 4, 267, 268, 271, 274, 301, 382, 625, 730]

[598] Анализируется путь [0, 3, 4, 267, 268, 271, 274, 301, 382, 625, 862, 880]

[599] Анализируется путь [0, 3, 4, 267, 268, 271, 274, 301, 382, 625, 862, 889]

[600] Анализируется путь [0, 3, 4, 267, 268, 271, 274, 301, 382, 625, 862, 943]

[601] Анализируется путь [0, 3, 4, 267, 268, 271, 274, 301, 382, 625, 862, 970]

[602] Циклов не найдено

[603]

[604] ----------

[605] Разборка графа

[606] [970] присвоен уровень 971

[607] [730, 889, 961, 969] присвоен уровень 970

[608] [490, 652, 721, 729, 811, 880, 888, 943, 952, 960, 967, 968] присвоен уровень 969

[609] [250, 409, 481, 489, 571, 643, 651, 703, 712, 720, 727, 728, 802, 810, 862, 871, 879, 886, 887, 916, 934, 942, 951, 958, 959, 964, 966] присвоен уровень 968

[610] [169, 241, 249, 328, 400, 408, 463, 472, 480, 487, 488, 562, 570, 625, 634, 642, 649, 650, 676, 694, 702, 711, 718, 719, 726, 784, 793, 801, 808, 809, 838, 854, 861, 870, 877, 878, 883, 885, 915, 925, 933, 940, 941, 949, 950, 955, 957, 963, 965] присвоен уровень 967

[611] [6, 88, 160, 168, 223, 232, 240, 247, 248, 319, 327, 382, 391, 399, 406, 407, 436, 454, 462, 471, 478, 479, 486, 544, 553, 561, 568, 569, 598, 616, 624, 633, 640, 641, 648, 675, 685, 693, 700, 701, 709, 710, 717, 724, 725, 757, 775, 783, 792, 799, 800, 805, 807, 837, 846, 853, 860, 868, 869, 874, 876, 882, 884, 907, 913, 914, 924, 931, 932, 937, 939, 946, 948, 954, 956, 962] присвоен уровень 966

[612] [7, 79, 87, 142, 151, 159, 166, 167, 196, 214, 222, 231, 238, 239, 246, 301, 310, 318, 325, 326, 355, 373, 381, 390, 397, 398, 405, 435, 445, 453, 460, 461, 469, 470, 477, 484, 485, 517, 535, 543, 552, 559, 560, 567, 597, 607, 615, 622, 623, 631, 632, 639, 646, 647, 668, 674, 684, 691, 692, 699, 708, 715, 716, 723, 756, 766, 774, 781, 782, 790, 791, 796, 798, 804, 806, 829, 835, 836, 845, 852, 857, 859, 865, 867, 873, 875, 881, 898, 906, 910, 912, 922, 923, 928, 930, 936, 938, 945, 947, 953] присвоен уровень 965

[613] [61, 70, 78, 85, 86, 115, 133, 141, 150, 157, 158, 165, 195, 205, 213, 220, 221, 229, 230, 237, 244, 245, 274, 292, 300, 309, 316, 317, 324, 354, 364, 372, 379, 380, 388, 389, 396, 403, 404, 427, 433, 434, 444, 451, 452, 459, 468, 475, 476, 483, 516, 526, 534, 541, 542, 550, 551, 558, 565, 566, 589, 595, 596, 606, 613, 614, 621, 630, 637, 638, 645, 660, 667, 673, 682, 683, 690, 697, 698, 706, 707, 714, 722, 748, 754, 755, 765, 772, 773, 778, 780, 787, 789, 795, 797, 803, 820, 828, 832, 834, 844, 849, 851, 856, 858, 864, 866, 872, 897, 904, 905, 909, 911, 919, 921, 927, 929, 935, 944] присвоен уровень 964

[614] [34, 52, 60, 69, 76, 77, 84, 114, 124, 132, 139, 140, 148, 149, 156, 163, 164, 187, 193, 194, 204, 211, 212, 219, 228, 235, 236, 243, 273, 283, 291, 298, 299, 307, 308, 315, 322, 323, 346, 352, 353, 363, 370, 371, 378, 387, 394, 395, 402, 418, 426, 432, 442, 443, 450, 457, 458, 466, 467, 474, 482, 508, 514, 515, 525, 532, 533, 540, 549, 556, 557, 564, 580, 588, 594, 604, 605, 612, 619, 620, 628, 629, 636, 644, 659, 666, 671, 672, 681, 688, 689, 696, 705, 713, 739, 747, 751, 753, 763, 764, 769, 771, 777, 779, 786, 788, 794, 819, 826, 827, 831, 833, 841, 843, 848, 850, 855, 863, 895, 896, 901, 903, 908, 918, 920, 926] присвоен уровень 963

[615] [33, 43, 51, 58, 59, 67, 68, 75, 82, 83, 106, 112, 113, 123, 130, 131, 138, 147, 154, 155, 162, 178, 186, 192, 202, 203, 210, 217, 218, 226, 227, 234, 242, 266, 271, 272, 282, 289, 290, 297, 306, 313, 314, 321, 337, 345, 351, 361, 362, 369, 376, 377, 385, 386, 393, 401, 417, 424, 425, 430, 431, 441, 448, 449, 456, 465, 473, 499, 507, 513, 523, 524, 531, 538, 539, 547, 548, 555, 563, 579, 586, 587, 592, 593, 603, 610, 611, 618, 627, 635, 658, 665, 670, 679, 680, 687, 695, 704, 738, 745, 746, 750, 752, 760, 762, 768, 770, 776, 785, 817, 818, 823, 825, 830, 840, 842, 847, 892, 894, 900, 902, 917] присвоен уровень 962

[616] [25, 31, 32, 42, 49, 50, 57, 66, 73, 74, 81, 97, 105, 111, 121, 122, 129, 136, 137, 145, 146, 153, 161, 177, 184, 185, 190, 191, 201, 208, 209, 216, 225, 233, 259, 265, 270, 280, 281, 288, 295, 296, 304, 305, 312, 320, 336, 343, 344, 349, 350, 360, 367, 368, 375, 384, 392, 415, 416, 423, 429, 439, 440, 447, 455, 464, 498, 505, 506, 511, 512, 522, 529, 530, 537, 546, 554, 577, 578, 585, 591, 601, 602, 609, 617, 626, 657, 663, 664, 669, 678, 686, 736, 737, 742, 744, 749, 759, 761, 767, 814, 816, 822, 824, 839, 891, 893, 899] присвоен уровень 961

[617] [16, 24, 30, 40, 41, 48, 55, 56, 64, 65, 72, 80, 96, 103, 104, 109, 110, 120, 127, 128, 135, 144, 152, 175, 176, 183, 189, 199, 200, 207, 215, 224, 258, 263, 264, 268, 269, 279, 286, 287, 294, 303, 311, 334, 335, 342, 348, 358, 359, 366, 374, 383, 414, 421, 422, 428, 438, 446, 496, 497, 504, 510, 520, 521, 528, 536, 545, 576, 583, 584, 590, 600, 608, 655, 656, 662, 677, 733, 735, 741, 743, 758, 813, 815, 821, 890] присвоен уровень 960

[618] [15, 22, 23, 28, 29, 39, 46, 47, 54, 63, 71, 94, 95, 102, 108, 118, 119, 126, 134, 143, 174, 181, 182, 188, 198, 206, 256, 257, 262, 267, 277, 278, 285, 293, 302, 333, 340, 341, 347, 357, 365, 412, 413, 420, 437, 495, 502, 503, 509, 519, 527, 574, 575, 582, 599, 654, 661, 732, 734, 740, 812] присвоен уровень 959

[619] [4, 5, 13, 14, 21, 27, 37, 38, 45, 53, 62, 93, 100, 101, 107, 117, 125, 172, 173, 180, 197, 255, 261, 276, 284, 331, 332, 339, 356, 411, 419, 493, 494, 501, 518, 573, 581, 653, 731] присвоен уровень 958

[620] [12, 19, 20, 26, 36, 44, 91, 92, 99, 116, 171, 179, 253, 254, 260, 275, 330, 338, 410, 492, 500, 572] присвоен уровень 957

[621] [3, 10, 11, 18, 35, 90, 98, 170, 252, 329, 491] присвоен уровень 956

[622] [0, 1, 2, 9, 17, 89, 251] присвоен уровень 955

[623] [8] присвоен уровень 954

[624]

[625] Разборка завершена

[626] Число уровней: 18

[627] Z = 98%

[628]

[629] ----------

[630] Проверка пересечений, которые могут появится после "разборки" графа

[631] пересечений не найдено

[632]

[633] ----------

[634] Опередедение границ областей для нормировки

[635] min=[[2, 0, 0, 0, 2, 0], [0, 0, 0, 0, 0, 0], [2, 0, 0, 0, 2, 0], [2, 0, 0, 0, 2, 0], [2, 0, 0, 0, 2, 0], [2, 0, 0, 0, 2, 0], [2, 0, 0, 0, 2, 0], [2, 0, 0, 0, 2, 0], [2, 0, 0, 0, 2, 0], [2, 0, 0, 0, 2, 0], [2, 0, 0, 0, 2, 0], [2, 0, 0, 0, 2, 0], [2, 0, 0, 0, 2, 0], [2, 0, 0, 0, 2, 1], [3, 0, 1, 0, 3, 2], [4, 0, 2, 0, 4, 3]]

[636] max=[[2, 155, 0, 0.2, 2, 0], [4, 256, 2, 1, 4, 3], [4, 173, 1, 0.4, 4, 2], [4, 173, 2, 0.4, 4, 2], [4, 256, 2, 1, 4, 3], [4, 256, 2, 1, 4, 3], [4, 256, 2, 1, 4, 3], [4, 256, 2, 1, 4, 3], [4, 256, 2, 1, 4, 3], [4, 256, 2, 1, 4, 3], [4, 256, 2, 1, 4, 3], [4, 256, 2, 1, 4, 3], [4, 256, 2, 1, 4, 3], [4, 256, 2, 1, 4, 3], [4, 256, 2, 1, 4, 3], [4, 256, 2, 1, 4, 3], [4, 256, 2, 1, 4, 3], [4, 256, 2, 1, 4, 3], [4, 256, 2, 1, 4, 3]]

[637]

[638] ----------

[639] Ранги альтернатив

[640] Акустика-М p = 1 d = 0.6212 y = 1.6212

[641] Альбедо p = 1 d = 0.8859 y = 1.8859

[642] АРИЛ p = 11 d = 0.6680 y = 11.6680

[643] Астра-3 p = 1 d = 0.6986 y = 1.6986

[644] Астровакцина p = 1 d = 0.5993 y = 1.5993

[645] Бактериофаг p = 1 d = 0.7400 y = 1.7400

[646] Биоген p = 6 d = 0.6052 y = 6.6052

[647] Биодеградация p = 11 d = 0.4709 y = 11.4709

[648] Биомагнистат p = 11 d = 0.3909 y = 11.3909

[649] Биотест p = 1 d = 0.1940 y = 1.1940

[650] Биотрек p = 1 d = 0.8478 y = 1.8478

[651] Биоэмульсия p = 1 d = 0.7344 y = 1.7344
[652] БИФ p = 4 d = 0.6746 y = 4.6746
[653] БСМК p = 1 d = 0.7003 y = 1.7003
[654] БТН-Нейтрон-2 p = 1 d = 0.2214 y = 1.2214
[655] Бюон p = 1 d = 0.4447 y = 1.4447
[656] Вектор-Т p = 1 d = 0.3838 y = 1.3838
[657] Великое начало p = 1 d = 0.4548 y = 1.4548
[658] Ветерок p = 1 d = 0.5134 y = 1.5134
[659] Вибролаб p = 3 d = 0.7815 y = 3.7815
[660] Визир p = 4 d = 0.3808 y = 4.3808
[661] ВИРУ p = 11 d = 0.4459 y = 11.4459
[662] Виток-2 p = 1 d = 0.2319 y = 1.2319
[663] Выносливость p = 1 d = 0.5942 y = 1.5942
[664] Гематология p = 1 d = 0.2093 y = 1.2093
[665] ГРИС – ФКИ – 1 p = 1 d = 0.5319 y = 1.5319
[666] Дальность p = 11 d = 0.3379 y = 11.3379
[667] Диурезр = 1 d = 0.2082 y = 1.2082
[668] Диффузия p = 11 d = 0.4051 y = 11.4051
[669] Дыхание p = 1 d = 0.1515 y = 1.1515
[670] Женьшень-2 p = 11 d = 0.7720 y = 11.7720
[671] Захват-Э p = 1 d = 0.6719 y = 1.6719
[672] Идентификация p = 1 d = 0.8805 y = 1.8805
[673] Изгиб p = 1 d = 0.8186 y = 1.8186
[674] Изменчивость p = 1 d = 0.7351 y = 1.7351
[675] Инфразвук-М p = 1 d = 0.6801 y = 1.6801
[676] Искажение p = 3 d = 0.6152 y = 3.6152
[677] Кальций p = 1 d = 0.6885 y = 1.6885
[678] Капля-2 p = 1 d = 0.2531 y = 1.2531
[679] КЛПВЭ p = 1 d = 0.7696 y = 1.7696
[680] Кожа p = 1 d = 0.6957 y = 1.6957
[681] Контур p = 1 d = 0.6511 y = 1.6511
[682] Контур-2 p = 1 d = 0.8227 y = 1.8227
[683] Конъюгация p = 1 d = 0.7458 y = 1.7458
[684] Криоконсервация p = 1 d = 0.6565 y = 1.6565
[685] Кромка p = 1 d = 0.7150 y = 1.7150

[686] Кулоновский кристалл p = 11 d = 0.4277 y = 11.4277
[687] Лактолен p = 11 d = 0.7074 y = 11.7074
[688] МАИ-75 p = 4 d = 0.2570 y = 4.2570
[689] МАТИ-75 p = 1 d = 0.4042 y = 1.4042
[690] Матрешка-Р p = 4 d = 0.5241 y = 4.5241
[691] МВН p = 6 d = 0.6507 y = 6.6507
[692] Межклеточное взаимодействие p = 1 d = 0.2781 y = 1.2781
[693] Мембрана p = 12 d = 0.7206 y = 12.7206
[694] Метаболизм p = 1 d = 0.5740 y = 1.5740
[695] МСК p = 1 d = 0.6262 y = 1.6262
[696] Наноспутник p = 6 d = 0.4394 y = 6.4394
[697] ОЧБ p = 11 d = 0.6412 y = 11.6412
[698] Перспектива-КМ p = 3 d = 0.8209 y = 3.8209
[699] Пилот p = 1 d = 0.1777 y = 1.1777
[700] Плазма-ЭРП p = 2 d = 0.4445 y = 2.4445
[701] Платан p = 11 d = 0.5653 y = 11.5653
[702] Пневмокард p = 1 d = 0.2613 y = 1.2613
[703] Полиген p = 1 d = 0.2844 y = 1.2844
[704] Привязка p = 11 d = 0.4849 y = 11.4849
[705] Пульс p = 1 d = 0.2329 y = 1.2329
[706] РадиоСкаф p = 3 d = 0.6087 y = 3.6087
[707] Регенерация p = 1 d = 0.1520 y = 1.1520
[708] Реставрация p = 1 d = 0.3260 y = 1.3260
[709] Рецептор p = 1 d = 0.2214 y = 1.2214
[710] Сейнер p = 1 d = 0.4023 y = 1.4023
[711] Секция ЭДК p = 1 d = 0.8584 y = 1.8584
[712] Сепарация p = 1 d = 0.6048 y = 1.6048
[713] Скорпион p = 11 d = 0.4427 y = 11.4427
[714] СЛС p = 1 d = 0.7687 y = 1.7687
[715] Сонокард p = 1 d = 0.2424 y = 1.2424
[716] Среда МКС p = 11 d = 0.4834 y = 11.4834
[717] Статокония p = 1 d = 0.2592 y = 1.2592
[718] Стресс p = 4 d = 0.4387 y = 4.4387
[719] Структура p = 11 d = 0.6425 y = 11.6425
[720] Тензор p = 11 d = 0.7678 y = 11.7678

[721] Тень-Маяк p = 11 d = 0.3372 y = 11.3372
[722] Типология p = 1 d = 0.1856 y = 1.1856
[723] Токсичность p = 2 d = 0.4731 y = 2.4731
[724] Трибокосмос p = 11 d = 0.7373 y = 11.7373
[725] Трос-МГТУ p = 1 d = 0.2418 y = 1.2418
[726] Фазопереход p = 1 d = 0.8690 y = 1.8690
[727] Фермент p = 12 d = 0.6181 y = 12.6181
[728] Физика-Образование p = 3 d = 0.5325 y = 3.5325
[729] Химия-Образование p = 4 d = 0.3027 y = 4.3027
[730] Эколинс p = 1 d = 0.7364 y = 1.7364
[731] Экология-Образование p = 11 d = 0.3473 y = 11.3473
[732] Эпсилон-НЭП p = 1 d = 0.7920 y = 1.7920
[733]
[734] ----------
[735] Ранжированный список альтернатив
[736] 1.Мембрана 12.7206
[737] 2.Фермент 12.6181
[738] 3.Женьшень-2 11.7720
[739] 4.Тензор 11.7678
[740] 5.Трибокосмос 11.7373
[741] 6.Лактолен 11.7074
[742] 7.АРИЛ 11.6680
[743] 8.Структура 11.6425
[744] 9.ОЧБ 11.6412
[745] 10.Платан 11.5653
[746] 11.Привязка 11.4849
[747] 12.Среда МКС 11.4834
[748] 13.Биодеградация 11.4709
[749] 14.ВИРУ 11.4459
[750] 15.Скорпион 11.4427
[751] 16.Кулоновский кристалл 11.4277
[752] 17.Диффузия 11.4051
[753] 18.Биомагнистат 11.3909
[754] 19.Экология-Образование 11.3473
[755] 20.Дальность 11.3379

[756] 21.Тень-Маяк 11.3372

[757] 22.МВН 6.6507

[758] 23.Биоген 6.6052

[759] 24.Наноспутник 6.4394

[760] 25.БИФ 4.6746

[761] 26.Матрешка-Р 4.5241

[762] 27.Стресс 4.4387

[763] 28.Визир 4.3808

[764] 29.Химия-Образование 4.3027

[765] 30.МАИ-75 4.2570

[766] 31.Перспектива-КМ 3.8209

[767] 32.Вибролаб 3.7815

[768] 33.Искажение 3.6152

[769] 34.РадиоСкаф 3.6087

[770] 35.Физика-Образование 3.5325

[771] 36.Токсичность 2.4731

[772] 37.Плазма-ЭРП 2.4445

[773] 38.Альбедо 1.8859

[774] 39.Идентификация 1.8805

[775] 40.Фазопереход 1.8690

[776] 41.Секция ЭДК 1.8584

[777] 42.Биотрек 1.8478

[778] 43.Контур-2 1.8227

[779] 44.Изгиб 1.8186

[780] 45.Эпсилон-НЭП 1.7920

[781] 46.КЛПВЭ 1.7696

[782] 47.СЛС 1.7687

[783] 48.Конъюгация 1.7458

[784] 49.Бактериофаг 1.7400

[785] 50.Эколинс 1.7364

[786] 51.Изменчивость 1.7351

[787] 52.Биоэмульсия 1.7344

[788] 53.Кромка 1.7150

[789] 54.БСМК 1.7003

[790] 55.Астра-3 1.6986

[791] 56.Кожа 1.6957

[792] 57.Кальций 1.6885

[793] 58.Инфразвук-М 1.6801

[794] 59.Захват-Э 1.6719

[795] 60.Криоконсервация 1.6565

[796] 61.Контур 1.6511

[797] 62.МСК 1.6262

[798] 63.Акустика-М 1.6212

[799] 64.Сепарация 1.6048

[800] 65.Астровакцина 1.5993

[801] 66.Выносливость 1.5942

[802] 67.Метаболизм 1.5740

[803] 68.ГРИС – ФКИ – 1 1.5319

[804] 69.Ветерок 1.5134

[805] 70.Великое начало 1.4548

[806] 71.Бюон 1.4447

[807] 72.МАТИ-75 1.4042

[808] 73.Сейнер 1.4023

[809] 74.Вектор-Т 1.3838

[810] 75.Реставрация 1.3260

[811] 76.Полиген 1.2844

[812] 77.Межклеточное взаимодействие 1.2781

[813] 78.Пневмокард 1.2613

[814] 79.Статокония 1.2592

[815] 80.Капля-2 1.2531

[816] 81.Сонокард 1.2424

[817] 82.Трос-МГТУ 1.2418

[818] 83.Пульс 1.2329

[819] 84.Виток-2 1.2319

[820] 85.БТН-Нейтрон-2 1.2214

[821] 86.Рецептор 1.2214

[822] 87.Гематология 1.2093

[823] 88.Диурез 1.2082

[824] 89.Биотест 1.1940

[825] 90.Типология 1.1856

[826] 91.Пилот 1.1777

[827] 92.Регенерация 1.1520

[828] 93.Дыхание 1.1515

Трассировка при наличии цикла

[829] ----------

[830] Исходные данные в формате JSON

[831] {

[832] "criterions": [

[833] {"name":"Критерий1", "scale": [0,1,2], "rank": 1},

[834] {"name":"Критерий2", "scale": [0,1,2], "rank": 2},

[835] {"name":"Критерий3", "scale": [0,1,2], "rank": 3}

[836]],

[837] "pref_scale": ["1", "2"],

[838] "preference": {

[839] "1":[

[840] {"Критерий1":[0], "Критерий2":[1], "Критерий3":[2]}

[841]],

[842] "2":[

[843] {"Критерий1":[2], "Критерий2":[0], "Критерий3":[0]}

[844]]

[845] },

[846] "alternative": {

[847] "Альтернатива 1":[1,1,1],

[848] "Альтернатива 2":[1,2,1]

[849]

[850] }

[851] }

[852]

[853] ----------

[854] Заполнение структур данных и проверка на пересечение множеств

[855]

[856] Множество № 0 Уровень предпочтений P = 0

[857] [[0], [1], [2]]

[858] пересечений не найдено

[859] Множество № 1 Уровень предпочтений P = 1

[860] [[2], [0], [0]]

[861] пересечений не найдено

[862]

[863] ----------

[864] Поиск ячеек в графе

[865]

[866] q = [3, 3, 3] Q = 27

[867] Добавлена ячейка T[0] = [0, 0, 0]

[868] Добавлена ячейка T[1] = [1, 0, 0]

[869] Добавлена ячейка T[2] = [0, 1, 0]

[870] Добавлена ячейка T[3] = [1, 1, 0]

[871] Добавлена ячейка T[4] = [2, 1, 0]

[872] Добавлена ячейка T[5] = [0, 2, 0]

[873] Добавлена ячейка T[6] = [1, 2, 0]

[874] Добавлена ячейка T[7] = [2, 2, 0]

[875] Добавлена ячейка T[8] = [0, 0, 1]

[876] Добавлена ячейка T[9] = [1, 0, 1]

[877] Добавлена ячейка T[10] = [2, 0, 1]

[878] Добавлена ячейка T[11] = [0, 1, 1]

[879] Добавлена ячейка T[12] = [1, 1, 1]

[880] Добавлена ячейка T[13] = [2, 1, 1]

[881] Добавлена ячейка T[14] = [0, 2, 1]

[882] Добавлена ячейка T[15] = [1, 2, 1]

[883] Добавлена ячейка T[16] = [2, 2, 1]

[884] Добавлена ячейка T[17] = [0, 0, 2]

[885] Добавлена ячейка T[18] = [1, 0, 2]

[886] Добавлена ячейка T[19] = [2, 0, 2]

[887] Добавлена ячейка T[20] = [1, 1, 2]

[888] Добавлена ячейка T[21] = [2, 1, 2]

[889] Добавлена ячейка T[22] = [0, 2, 2]

[890] Добавлена ячейка T[23] = [1, 2, 2]

[891] Добавлена ячейка T[24] = [2, 2, 2]

[892] Всего множеств и ячеек: 27

[893]

[894] ----------

[895] Определение отношений доминирования по Парето и по Подиновскому

[896]

[897] [[0], [1], [2]] доминирует [[0], [0], [0]] по Парето

[898] [[0], [1], [2]] доминирует [[0], [1], [0]] по Парето

[899] [[0], [1], [2]] доминирует [[2], [1], [0]] по Подиновскому

[900] [[0], [1], [2]] доминирует [[0], [0], [1]] по Парето

[901] [[0], [1], [2]] доминирует [[0], [1], [1]] по Парето

[902] [[0], [1], [2]] доминирует [[0], [2], [1]] по Подиновскому

[903] [[0], [1], [2]] доминирует [[0], [0], [2]] по Парето

[904] [[0], [1], [2]] доминирует [[1], [0], [2]] по Подиновскому

[905] [[2], [0], [0]] доминирует [[0], [0], [0]] по Парето

[906] [[2], [0], [0]] доминирует [[1], [0], [0]] по Парето

[907] [[1], [0], [0]] доминирует [[0], [0], [0]] по Парето

[908] [[0], [1], [0]] доминирует [[0], [0], [0]] по Парето

[909] [[0], [1], [0]] доминирует [[1], [0], [0]] по Подиновскому

[910] [[1], [1], [0]] доминирует [[0], [0], [0]] по Парето

[911] [[1], [1], [0]] доминирует [[1], [0], [0]] по Парето

[912] [[1], [1], [0]] доминирует [[0], [1], [0]] по Парето

[913] [[2], [1], [0]] доминирует [[2], [0], [0]] по Парето

[914] [[2], [1], [0]] доминирует [[0], [0], [0]] по Парето

[915] [[2], [1], [0]] доминирует [[1], [0], [0]] по Парето

[916] [[2], [1], [0]] доминирует [[0], [1], [0]] по Парето

[917] [[2], [1], [0]] доминирует [[1], [1], [0]] по Парето

[918] [[0], [2], [0]] доминирует [[2], [0], [0]] по Подиновскому

[919] [[0], [2], [0]] доминирует [[0], [0], [0]] по Парето

[920] [[0], [2], [0]] доминирует [[0], [1], [0]] по Парето

[921] [[1], [2], [0]] доминирует [[0], [0], [0]] по Парето

[922] [[1], [2], [0]] доминирует [[1], [0], [0]] по Парето

[923] [[1], [2], [0]] доминирует [[0], [1], [0]] по Парето

[924] [[1], [2], [0]] доминирует [[1], [1], [0]] по Парето

[925] [[1], [2], [0]] доминирует [[2], [1], [0]] по Подиновскому

[926] [[1], [2], [0]] доминирует [[0], [2], [0]] по Парето

[927] [[2], [2], [0]] доминирует [[2], [0], [0]] по Парето

[928] [[2], [2], [0]] доминирует [[0], [0], [0]] по Парето

[929] [[2], [2], [0]] доминирует [[1], [0], [0]] по Парето

[930] [[2], [2], [0]] доминирует [[0], [1], [0]] по Парето

[931] [[2], [2], [0]] доминирует [[1], [1], [0]] по Парето

[932] [[2], [2], [0]] доминирует [[2], [1], [0]] по Парето

[933] [[2], [2], [0]] доминирует [[0], [2], [0]] по Парето

[934] [[2], [2], [0]] доминирует [[1], [2], [0]] по Парето

[935] [[0], [0], [1]] доминирует [[0], [0], [0]] по Парето

[936] [[0], [0], [1]] доминирует [[1], [0], [0]] по Подиновскому

[937] [[0], [0], [1]] доминирует [[0], [1], [0]] по Подиновскому

[938] [[1], [0], [1]] доминирует [[0], [0], [0]] по Парето

[939] [[1], [0], [1]] доминирует [[1], [0], [0]] по Парето

[940] [[1], [0], [1]] доминирует [[1], [1], [0]] по Подиновскому

[941] [[1], [0], [1]] доминирует [[0], [0], [1]] по Парето

[942] [[2], [0], [1]] доминирует [[2], [0], [0]] по Парето

[943] [[2], [0], [1]] доминирует [[0], [0], [0]] по Парето

[944] [[2], [0], [1]] доминирует [[1], [0], [0]] по Парето

[945] [[2], [0], [1]] доминирует [[2], [1], [0]] по Подиновскому

[946] [[2], [0], [1]] доминирует [[0], [0], [1]] по Парето

[947] [[2], [0], [1]] доминирует [[1], [0], [1]] по Парето

[948] [[0], [1], [1]] доминирует [[0], [0], [0]] по Парето

[949] [[0], [1], [1]] доминирует [[0], [1], [0]] по Парето

[950] [[0], [1], [1]] доминирует [[1], [1], [0]] по Подиновскому

[951] [[0], [1], [1]] доминирует [[0], [0], [1]] по Парето

[952] [[0], [1], [1]] доминирует [[1], [0], [1]] по Подиновскому

[953] [[1], [1], [1]] доминирует [[0], [0], [0]] по Парето

[954] [[1], [1], [1]] доминирует [[1], [0], [0]] по Парето

[955] [[1], [1], [1]] доминирует [[0], [1], [0]] по Парето

[956] [[1], [1], [1]] доминирует [[1], [1], [0]] по Парето

[957] [[1], [1], [1]] доминирует [[0], [0], [1]] по Парето

[958] [[1], [1], [1]] доминирует [[1], [0], [1]] по Парето

[959] [[1], [1], [1]] доминирует [[0], [1], [1]] по Парето

[960] [[2], [1], [1]] доминирует [[2], [0], [0]] по Парето

[961] [[2], [1], [1]] доминирует [[0], [0], [0]] по Парето

[962] [[2], [1], [1]] доминирует [[1], [0], [0]] по Парето

[963] [[2], [1], [1]] доминирует [[0], [1], [0]] по Парето

[964] [[2], [1], [1]] доминирует [[1], [1], [0]] по Парето

[965] [[2], [1], [1]] доминирует [[2], [1], [0]] по Парето

[966] [[2], [1], [1]] доминирует [[0], [0], [1]] по Парето

[967] [[2], [1], [1]] доминирует [[1], [0], [1]] по Парето

[968] [[2], [1], [1]] доминирует [[2], [0], [1]] по Парето

[969] [[2], [1], [1]] доминирует [[0], [1], [1]] по Парето

[970] [[2], [1], [1]] доминирует [[1], [1], [1]] по Парето

[971] [[0], [2], [1]] доминирует [[0], [0], [0]] по Парето

[972] [[0], [2], [1]] доминирует [[0], [1], [0]] по Парето

[973] [[0], [2], [1]] доминирует [[0], [2], [0]] по Парето

[974] [[0], [2], [1]] доминирует [[1], [2], [0]] по Подиновскому

[975] [[0], [2], [1]] доминирует [[0], [0], [1]] по Парето

[976] [[0], [2], [1]] доминирует [[2], [0], [1]] по Подиновскому

[977] [[0], [2], [1]] доминирует [[0], [1], [1]] по Парето

[978] [[1], [2], [1]] доминирует [[0], [0], [0]] по Парето

[979] [[1], [2], [1]] доминирует [[1], [0], [0]] по Парето

[980] [[1], [2], [1]] доминирует [[0], [1], [0]] по Парето

[981] [[1], [2], [1]] доминирует [[1], [1], [0]] по Парето

[982] [[1], [2], [1]] доминирует [[0], [2], [0]] по Парето

[983] [[1], [2], [1]] доминирует [[1], [2], [0]] по Парето

[984] [[1], [2], [1]] доминирует [[0], [0], [1]] по Парето

[985] [[1], [2], [1]] доминирует [[1], [0], [1]] по Парето

[986] [[1], [2], [1]] доминирует [[0], [1], [1]] по Парето

[987] [[1], [2], [1]] доминирует [[1], [1], [1]] по Парето

[988] [[1], [2], [1]] доминирует [[2], [1], [1]] по Подиновскому

[989] [[1], [2], [1]] доминирует [[0], [2], [1]] по Парето

[990] [[2], [2], [1]] доминирует [[2], [0], [0]] по Парето

[991] [[2], [2], [1]] доминирует [[0], [0], [0]] по Парето

[992] [[2], [2], [1]] доминирует [[1], [0], [0]] по Парето

[993] [[2], [2], [1]] доминирует [[0], [1], [0]] по Парето

[994] [[2], [2], [1]] доминирует [[1], [1], [0]] по Парето

[995] [[2], [2], [1]] доминирует [[2], [1], [0]] по Парето

[996] [[2], [2], [1]] доминирует [[0], [2], [0]] по Парето

[997] [[2], [2], [1]] доминирует [[1], [2], [0]] по Парето

[998] [[2], [2], [1]] доминирует [[2], [2], [0]] по Парето

[999] [[2], [2], [1]] доминирует [[0], [0], [1]] по Парето

[1000] [[2], [2], [1]] доминирует [[1], [0], [1]] по Парето

[1001] [[2], [2], [1]] доминирует [[2], [0], [1]] по Парето

[1002] [[2], [2], [1]] доминирует [[0], [1], [1]] по Парето

[1003] [[2], [2], [1]] доминирует [[1], [1], [1]] по Парето

[1004] [[2], [2], [1]] доминирует [[2], [1], [1]] по Парето

[1005] [[2], [2], [1]] доминирует [[0], [2], [1]] по Парето

[1006] [[2], [2], [1]] доминирует [[1], [2], [1]] по Парето

[1007] [[0], [0], [2]] доминирует [[2], [0], [0]] по Подиновскому

[1008] [[0], [0], [2]] доминирует [[0], [0], [0]] по Парето

[1009] [[0], [0], [2]] доминирует [[0], [2], [0]] по Подиновскому

[1010] [[0], [0], [2]] доминирует [[0], [0], [1]] по Парето

[1011] [[1], [0], [2]] доминирует [[0], [0], [0]] по Парето

[1012] [[1], [0], [2]] доминирует [[1], [0], [0]] по Парето

[1013] [[1], [0], [2]] доминирует [[1], [2], [0]] по Подиновскому

[1014] [[1], [0], [2]] доминирует [[2], [0], [1]] по Подиновскому

[1015] [[2], [0], [2]] доминирует [[2], [2], [0]] по Подиновскому

[1016] [[1], [1], [2]] доминирует [[2], [1], [1]] по Подиновскому

[1017] [[1], [1], [2]] доминирует [[1], [2], [1]] по Подиновскому

[1018] [[2], [1], [2]] доминирует [[2], [2], [1]] по Подиновскому

[1019] [[0], [2], [2]] доминирует [[2], [2], [0]] по Подиновскому

[1020] [[0], [2], [2]] доминирует [[2], [0], [2]] по Подиновскому

[1021] [[1], [2], [2]] доминирует [[2], [2], [1]] по Подиновскому

[1022] [[1], [2], [2]] доминирует [[2], [1], [2]] по Подиновскому

[1023] .....

[1024] Отношений доминирования по Парето: 189, по Подиновскому 27

[1025] Дополним граф отношениями доминирования, введенными пользователем

[1026] [[2], [0], [0]] доминирует [[0], [1], [2]]

[1027] Отношений доминирования по Парето: 189, по Подиновскому 27, пользовательских 1

[1028]

[1029] 26 Недоминируемая вершина: [[2], [2], [2]]

[1030] Всего дуг в графе доминирования: 217

[1031]

[1032] ----------

[1033] Поиск циклов

[1034] Анализируется путь [0, 22, 23, 25, 26]

[1035] Анализируется путь [0, 22, 23, 26]

[1036] Анализируется путь [0, 22, 25]

[1037] Анализируется путь [0, 22, 26]

[1038] Анализируется путь [0, 23]

[1039] Анализируется путь [0, 24, 25]

[1040] Анализируется путь [0, 24, 26]

[1041] Анализируется путь [0, 25]

[1042] Анализируется путь [0, 26]

[1043] Анализируется путь [0, 1, 6, 0]

[1044] Ошибка! Найден цикл [[0], [1], [2]](!)<-[[2], [0], [0]](!)<-[[2], [1], [0]]<-[[0], [1], [2]](!)<- нужно изменить уровни предпочтений или границы для множеств отмеченных знаком (!)

Приложение 2. Трассировка процедуры дискретизации шкалы

labda[2,ksi]	ksi	u[1,ksi]	labda[1,ksi]	u[2,ksi]	u[3,ksi]	labda[3,ksi]	u[4,ksi]	labda[4,ksi]
0.0000	1	1	0.0000	1	1	0.0000	1	0.0000
0.0000	2	2	0.0000	2	2	0.0000	2	0.0000
0.0000	3	3	0.0000	3	3	0.0000	3	0.0000
0.0000	4	4	0.0000	4	4	0.0000	4	0.0000
0.0000	5	5	0.0000	5	5	0.0000	5	0.0000
0.0000	6	6	0.0000	6	6	0.0000	6	0.0000
0.0000	7	7	0.0000	7	7	0.0000	7	0.0000
0.0000	8	8	0.0000	8	8	0.0000	8	0.0000
0.0000	9	9	0.0000	9	9	0.0000	9	0.0000
0.0000	10	10	0.0000	10	10	0.0000	10	0.0000
0.0000	11	11	0.0000	11	11	0.0000	11	0.0000
0.0000	12	12	0.0000	12	12	0.0000	12	0.0000
0.0000	13	13	0.0000	13	13	0.0000	13	0.0000
0.0000	14	14	0.0000	14	14	0.0000	14	0.0000
0.0000	15	15	0.0000	15	15	0.0000	15	0.0000
0.0000	16	16	0.0000	16	16	0.0000	16	0.0000
0.0000	17	17	0.0000	17	17	0.0000	17	0.0000
0.0000	18	18	0.0000	18	18	0.0000	18	0.0000
0.0000	19	19	0.0000	19	19	0.0000	19	0.0000
0.0000	20	20	0.0000	20	20	0.0000	20	0.0000
0.0000	21	21	0.0000	21	21	0.0000	21	0.0000
0.0000	22	22	0.0000	22	22	0.0000	22	0.0000
0.0000	23	23	0.0000	23	23	0.0000	23	0.0000
0.0000	24	24	0.0000	24	24	0.0000	24	0.0000

labda[2,ksi]	ksi	u[1,ksi]	labda[1,ksi]	u[2,ksi]	u[3,ksi]	labda[3,ksi]	u[4,ksi]	labda[4,ksi]
0.0000	25	25	0.0000	25	25	0.0000	25	0.0000
0.0000	26	26	0.0000	26	26	0.0000	26	0.0000
0.0000	27	27	0.0000	27	27	0.0000	27	0.0000
0.0000	28	28	0.0000	28	28	0.0000	28	0.0000
0.0000	29	29	0.0000	29	29	0.0000	29	0.0000
0.1191	30	30	0.1191	30	30	0.1191	30	0.1191
0.1191	31	31	0.1191	31	31	0.1191	31	0.1191
0.1191	32	32	0.1191	32	32	0.1191	32	0.1191
0.1191	33	33	0.1191	33	33	0.1191	33	0.1191
0.1191	34	34	0.1191	34	34	0.1191	34	0.1191
0.1191	35	35	0.1191	35	35	0.1191	35	0.1191
0.1191	36	36	0.1191	36	36	0.1191	36	0.1191
0.1191	37	37	0.1191	37	37	0.1191	37	0.1191
0.1191	38	38	0.1191	38	38	0.1191	38	0.1191
0.1191	39	39	0.1191	39	39	0.1191	39	0.1191
0.1191	40	40	0.1191	40	40	0.1191	40	0.1191
0.1191	41	41	0.1191	41	41	0.1191	41	0.1191
0.1191	42	42	0.1191	42	42	0.1191	42	0.1191
0.1191	43	43	0.1191	43	43	0.1191	43	0.1191
0.1191	44	44	0.1191	44	44	0.1191	44	0.1191
0.1894	45	45	0.1598	15	15	0.1894	15	0.1894
0.1894	46	46	0.1598	16	16	0.1894	16	0.1894
0.1894	47	47	0.1598	17	17	0.1894	17	0.1894
0.1894	48	48	0.1598	18	18	0.1894	18	0.1894
0.1894	49	49	0.1598	19	19	0.1894	19	0.1894
0.1894	50	50	0.1598	20	20	0.1894	20	0.1894
0.1894	51	51	0.1598	21	21	0.1894	21	0.1894
0.1894	52	52	0.1598	22	22	0.1894	22	0.1894
0.1894	53	53	0.1598	23	23	0.1894	23	0.1894
0.1894	54	54	0.1598	24	24	0.1894	24	0.1894
0.1894	55	55	0.1598	25	25	0.1894	25	0.1894
0.1894	56	56	0.1598	26	26	0.1894	26	0.1894
0.1894	57	57	0.1598	27	27	0.1894	27	0.1894
0.1894	58	58	0.1598	28	28	0.1894	28	0.1894
0.5566	59	59	0.4368	14	14	0.5862	14	0.5862
0.5566	60	60	0.4368	15	15	0.5862	15	0.5862
0.5566	61	61	0.4368	16	16	0.5862	16	0.5862
0.5566	62	62	0.4368	17	17	0.5862	17	0.5862
0.5566	63	63	0.4368	18	18	0.5862	18	0.5862
0.5566	64	64	0.4368	19	19	0.5862	19	0.5862
0.5566	65	65	0.4368	20	20	0.5862	20	0.5862
0.5566	66	66	0.4368	21	21	0.5862	21	0.5862
0.5566	67	67	0.4368	22	22	0.5862	22	0.5862
0.5566	68	68	0.4368	23	23	0.5862	23	0.5862

labda[2,ksi]	ksi	u[1,ksi]	labda[1,ksi]	u[2,ksi]	u[3,ksi]	labda[3,ksi]	u[4,ksi]	labda[4,ksi]
0.5566	69	69	0.4368	24	24	0.5862	24	0.5862
0.5566	70	70	0.4368	25	25	0.5862	25	0.5862
0.5566	71	71	0.4368	26	26	0.5862	26	0.5862
0.5566	72	72	0.4368	27	27	0.5862	27	0.5862
0.5566	73	73	0.4368	28	28	0.5862	28	0.5862
0.7177	74	74	0.4985	15	15	0.8375	15	0.8671
0.7177	75	75	0.4985	16	16	0.8375	16	0.8671
0.7177	76	76	0.4985	17	17	0.8375	17	0.8671
0.7177	77	77	0.4985	18	18	0.8375	18	0.8671
0.7177	78	78	0.4985	19	19	0.8375	19	0.8671
0.7177	79	79	0.4985	20	20	0.8375	20	0.8671
0.7177	80	80	0.4985	21	21	0.8375	21	0.8671
0.7177	81	81	0.4985	22	22	0.8375	22	0.8671
0.7177	82	82	0.4985	23	23	0.8375	23	0.8671
0.7177	83	83	0.4985	24	24	0.8375	24	0.8671
0.7177	84	84	0.4985	25	25	0.8375	25	0.8671
0.7177	85	85	0.4985	26	26	0.8375	26	0.8671
0.7177	86	86	0.4985	27	27	0.8375	27	0.8671
0.7177	87	87	0.4985	28	28	0.8375	28	0.8671
0.7177	88	88	0.4985	29	29	0.8375	29	0.8671
0.7177	89	89	0.4985	30	30	0.8375	30	0.8671
0.7177	90	90	0.4985	31	31	0.8375	31	0.8671
0.7177	91	91	0.4985	32	32	0.8375	32	0.8671
0.7177	92	92	0.4985	33	33	0.8375	33	0.8671
0.7177	93	93	0.4985	34	34	0.8375	34	0.8671
0.7177	94	94	0.4985	35	35	0.8375	35	0.8671
0.7177	95	95	0.4985	36	36	0.8375	36	0.8671
0.7177	96	96	0.4985	37	37	0.8375	37	0.8671
0.7177	97	97	0.4985	38	38	0.8375	38	0.8671
0.7413	98	98	0.5043	39	39	0.8611	24	0.9078
0.7413	99	99	0.5043	40	40	0.8611	25	0.9078
0.7413	100	100	0.5043	41	41	0.8611	26	0.9078
0.7413	101	101	0.5043	42	42	0.8611	27	0.9078
0.7629	102	102	0.5095	43	43	0.8827	28	0.9566
0.7629	103	103	0.5095	44	44	0.8827	29	0.9566
0.7828	104	104	0.5141	45	45	0.9026	30	0.9973
0.7828	105	105	0.5141	46	46	0.9026	31	0.9973
0.7828	106	106	0.5141	47	47	0.9026	32	0.9973
0.7828	107	107	0.5141	48	48	0.9026	33	0.9973
0.7828	108	108	0.5141	49	49	0.9026	34	0.9973
0.7828	109	109	0.5141	50	50	0.9026	35	0.9973
0.7828	110	110	0.5141	51	51	0.9026	36	0.9973
0.7828	111	111	0.5141	52	52	0.9026	37	0.9973
0.7828	112	112	0.5141	53	53	0.9026	38	0.9973

labda[2,ksi]	ksi	u[1,ksi]	labda[1,ksi]	u[2,ksi]	u[3,ksi]	labda[3,ksi]	u[4,ksi]	labda[4,ksi]
0.8011	113	113	0.5180	54	54	0.9209	39	1.0327
0.8011	114	114	0.5180	55	55	0.9209	40	1.0327
0.8011	115	115	0.5180	56	56	0.9209	41	1.0327
0.8481	116	116	0.5265	57	42	0.9986	42	1.1184
0.8614	117	117	0.5283	58	19	1.0222	19	1.1420
0.8737	118	118	0.5296	59	20	1.0457	20	1.1655
0.8737	119	119	0.5296	60	21	1.0457	21	1.1655
0.8737	120	120	0.5296	61	22	1.0457	22	1.1655
0.8737	121	121	0.5296	62	23	1.0457	23	1.1655
0.8737	122	122	0.5296	63	24	1.0457	24	1.1655
0.8737	123	123	0.5296	64	25	1.0457	25	1.1655
0.8737	124	124	0.5296	65	26	1.0457	26	1.1655
0.8737	125	125	0.5296	66	27	1.0457	27	1.1655
0.8737	126	126	0.5296	67	28	1.0457	28	1.1655
0.8850	127	127	0.5304	68	29	1.0673	29	1.1871
0.8954	128	128	0.5307	69	26	1.0890	26	1.2087
0.8954	129	129	0.5307	70	27	1.0890	27	1.2087
0.8954	130	130	0.5307	71	28	1.0890	28	1.2087
0.8954	131	131	0.5307	72	29	1.0890	29	1.2087
0.8954	132	132	0.5307	73	30	1.0890	30	1.2087
0.9049	133	133	0.5306	74	29	1.1088	31	1.2286
0.9049	134	134	0.5306	75	30	1.1088	32	1.2286
0.9049	135	135	0.5306	76	31	1.1088	33	1.2286
0.9049	136	136	0.5306	77	32	1.1088	34	1.2286
0.9049	137	137	0.5306	78	33	1.1088	35	1.2286
0.9049	138	138	0.5306	79	34	1.1088	36	1.2286
0.9049	139	139	0.5306	80	35	1.1088	37	1.2286
0.9049	140	140	0.5306	81	36	1.1088	38	1.2286
0.9049	141	141	0.5306	82	37	1.1088	39	1.2286
0.9216	142	97	0.5307	83	38	1.1471	26	1.2795
0.9833	143	84	0.5214	69	27	1.2593	10	1.4349
0.9833	144	85	0.5214	70	28	1.2593	11	1.4349
0.9833	145	86	0.5214	71	29	1.2593	12	1.4349
0.9833	146	87	0.5214	72	30	1.2593	13	1.4349
0.9833	147	88	0.5214	73	31	1.2593	14	1.4349
0.9905	148	89	0.5242	74	32	1.2726	15	1.4548
0.9905	149	90	0.5242	75	33	1.2726	16	1.4548
0.9905	150	91	0.5242	76	34	1.2726	17	1.4548
0.9905	151	92	0.5242	77	35	1.2726	18	1.4548
0.9905	152	93	0.5242	78	36	1.2726	19	1.4548
0.9905	153	94	0.5242	79	37	1.2726	20	1.4548
0.9905	154	95	0.5242	80	38	1.2726	21	1.4548
1.0086	155	96	0.5296	57	37	1.3105	13	1.5113
1.0086	156	97	0.5296	58	38	1.3105	14	1.5113

labda[2,ksi]	ksi	u[1,ksi]	labda[1,ksi]	u[2,ksi]	u[3,ksi]	labda[3,ksi]	u[4,ksi]	labda[4,ksi]
1.0190	157	98	0.5307	55	30	1.3332	15	1.5439
1.0446	158	84	0.5307	45	16	1.3984	15	1.6561
1.0446	159	85	0.5307	46	17	1.3984	16	1.6561
1.0549	160	47	0.5307	44	17	1.4314	12	1.7095
1.0549	161	48	0.5307	45	18	1.4314	13	1.7095
1.0549	162	49	0.5307	46	19	1.4314	14	1.7095
1.0566	163	50	0.5306	46	20	1.4418	8	1.7218
1.0566	164	51	0.5306	47	21	1.4418	9	1.7218
1.0566	165	52	0.5306	48	22	1.4418	10	1.7218
1.0566	166	53	0.5306	49	23	1.4418	11	1.7218
1.0566	167	54	0.5306	50	24	1.4418	12	1.7218
1.0566	168	55	0.5306	51	25	1.4418	13	1.7218
1.0566	169	56	0.5306	52	26	1.4418	14	1.7218
1.0566	170	57	0.5306	53	27	1.4418	15	1.7218
1.0566	171	58	0.5306	54	28	1.4418	16	1.7218
1.0592	172	56	0.5307	54	29	1.4601	17	1.7474
1.0612	173	40	0.5306	40	18	1.5006	16	1.8100
1.0612	174	41	0.5306	41	19	1.5006	17	1.8100
1.0615	175	33	0.5307	33	20	1.5071	18	1.8179
1.0613	176	34	0.5306	34	21	1.5129	19	1.8251
1.0583	177	34	0.5283	35	20	1.5285	19	1.8570
1.0583	178	35	0.5283	36	21	1.5285	20	1.8570
1.0583	179	36	0.5283	37	22	1.5285	21	1.8570
1.0583	180	37	0.5283	38	23	1.5285	22	1.8570
1.0583	181	38	0.5283	39	24	1.5285	23	1.8570
1.0583	182	39	0.5283	40	25	1.5285	24	1.8570
1.0583	183	40	0.5283	41	26	1.5285	25	1.8570
1.0583	184	41	0.5283	42	27	1.5285	26	1.8570
1.0583	185	42	0.5283	43	28	1.5285	27	1.8570
1.0565	186	43	0.5296	44	29	1.5330	28	1.8665
1.0592	187	39	0.5306	32	30	1.5455	24	1.9004
1.0592	188	40	0.5306	33	31	1.5455	25	1.9004
1.0592	189	41	0.5306	34	32	1.5455	26	1.9004
1.0592	190	42	0.5306	35	33	1.5455	27	1.9004
1.0592	191	43	0.5306	36	34	1.5455	28	1.9004
1.0592	192	44	0.5306	37	35	1.5455	29	1.9004
1.0592	193	45	0.5306	38	36	1.5455	30	1.9004
1.0592	194	46	0.5306	39	37	1.5455	31	1.9004
1.0592	195	47	0.5306	40	38	1.5455	32	1.9004
1.0592	196	48	0.5306	41	39	1.5455	33	1.9004
1.0592	197	49	0.5306	42	40	1.5455	34	1.9004
1.0592	198	50	0.5306	43	41	1.5455	35	1.9004
1.0603	199	44	0.5307	44	41	1.5540	36	1.9186
1.0611	200	45	0.5306	43	42	1.5586	28	1.9282

labda[2,ksi]	ksi	u[1,ksi]	labda[1,ksi]	u[2,ksi]	u[3,ksi]	labda[3,ksi]	u[4,ksi]	labda[4,ksi]
1.0598	201	44	0.5290	44	43	1.5711	29	1.9586
1.0590	202	44	0.5283	44	44	1.5729	30	1.9644
1.0590	203	45	0.5283	45	45	1.5729	31	1.9644
1.0590	204	46	0.5283	46	46	1.5729	32	1.9644
1.0590	205	47	0.5283	47	47	1.5729	33	1.9644
1.0590	206	48	0.5283	48	48	1.5729	34	1.9644
1.0590	207	49	0.5283	49	49	1.5729	35	1.9644
1.0590	208	50	0.5283	50	50	1.5729	36	1.9644
1.0590	209	51	0.5283	51	51	1.5729	37	1.9644
1.0590	210	52	0.5283	52	52	1.5729	38	1.9644
1.0590	211	53	0.5283	53	53	1.5729	39	1.9644
1.0590	212	54	0.5283	54	54	1.5729	40	1.9644
1.0590	213	55	0.5283	55	55	1.5729	41	1.9644
1.0603	214	56	0.5296	56	54	1.5763	42	1.9696
1.0611	215	57	0.5304	57	55	1.5791	42	1.9774
1.0611	216	58	0.5304	58	56	1.5791	43	1.9774
1.0615	217	59	0.5307	59	57	1.5814	44	1.9854
1.0615	218	60	0.5307	60	58	1.5814	45	1.9854
1.0615	219	61	0.5307	61	59	1.5814	46	1.9854
1.0615	220	62	0.5307	62	60	1.5814	47	1.9854
1.0615	221	63	0.5307	63	61	1.5814	48	1.9854
1.0615	222	64	0.5307	64	62	1.5814	49	1.9854
1.0615	223	65	0.5307	65	63	1.5814	50	1.9854
1.0615	224	66	0.5307	66	64	1.5814	51	1.9854
1.0615	225	67	0.5307	67	65	1.5814	52	1.9854
1.0615	226	68	0.5307	68	66	1.5814	53	1.9854
1.0615	227	69	0.5307	69	67	1.5814	54	1.9854
1.0615	228	70	0.5307	70	68	1.5814	55	1.9854
1.0615	229	71	0.5307	71	69	1.5814	56	1.9854
1.0613	230	72	0.5306	72	70	1.5832	57	1.9926
1.0613	231	73	0.5306	73	71	1.5832	58	1.9926
1.0613	232	74	0.5306	74	72	1.5832	59	1.9926
1.0613	233	75	0.5306	75	73	1.5832	60	1.9926
1.0613	234	76	0.5306	76	74	1.5832	61	1.9926
1.0613	235	77	0.5306	77	75	1.5832	62	1.9926
1.0613	236	78	0.5306	78	76	1.5832	63	1.9926
1.0613	237	79	0.5306	79	77	1.5832	64	1.9926
1.0613	238	80	0.5306	80	78	1.5832	65	1.9926
1.0613	239	81	0.5306	81	79	1.5832	66	1.9926
1.0613	240	82	0.5306	82	80	1.5832	67	1.9926
1.0613	241	83	0.5306	83	81	1.5832	68	1.9926
1.0613	242	84	0.5306	84	82	1.5832	69	1.9926
1.0613	243	85	0.5306	85	83	1.5832	70	1.9926
1.0613	244	86	0.5306	86	84	1.5832	71	1.9926

labda[2,ksi]	ksi	u[1,ksi]	labda[1,ksi]	u[2,ksi]	u[3,ksi]	labda[3,ksi]	u[4,ksi]	labda[4,ksi]
1.0613	245	87	0.5306	87	85	1.5832	72	1.9926
1.0613	246	88	0.5306	88	86	1.5832	73	1.9926
1.0613	247	89	0.5306	89	87	1.5832	74	1.9926
1.0613	248	90	0.5306	90	88	1.5832	75	1.9926
1.0613	249	91	0.5306	91	89	1.5832	76	1.9926
1.0613	250	92	0.5306	92	90	1.5832	77	1.9926
1.0613	251	93	0.5306	93	91	1.5832	78	1.9926
1.0613	252	94	0.5306	94	92	1.5832	79	1.9926
1.0613	253	95	0.5306	95	93	1.5832	80	1.9926
1.0613	254	96	0.5306	96	94	1.5832	81	1.9926
1.0613	255	97	0.5306	97	95	1.5832	82	1.9926
1.0608	256	98	0.5300	98	93	1.5850	83	1.9991

Найденное оптимальное решение

98 57 18 83

Приложение 3. Внедрение результатов диссертационной работы

"УТВЕРЖДАЮ"
Проректор Института
_____ В.Г.Веретенников
"___" _____ 200_ г.
М.П.

"УТВЕРЖДАЮ"
Генеральный директор
ФГУП НПО "ОРИОН"
_____ А. С. Демидов
"___" _____ 200_ г.
М.П.

"СОГЛАСОВАНО"
Начальник 968 ВП МО РФ
_____ В.Г.Каленов
"___" _____ 200_ г.
М.П.

ТЕХНИЧЕСКИЙ АКТ 178/1 от 15.05.03
о выполнении работ
по договору № 22560-03020 от 3.01.2002 г.

Мы, ниже подписавшиеся, представитель Исполнителя – проректор по научной работе Московского авиационного института (Государственного Технического Университета) В.Г.ВЕРЕТЕННИКОВ с одной стороны, и представитель Заказчика Генеральный директор ФГУП НПО "Орион" А.С.Демидов с другой стороны, составили настоящий акт о том, что работа **«Разработка программных комплексов обеспечения деятельности войск»** по договору № 22560-03020 от 3.01.2002 г. выполнена в полном объеме и в надлежащем порядке оформлена.

Отчетные материалы.

1. Итоговый отчет, включающий:
   - общее описание системы;
   - руководство пользователя.
2. Программный продукт на электронном носителе.

Договорная цена работы по договору составляет **600 000,0** (шестьсот тысяч) рублей без НДС.
Сумма аванса, перечисленная Заказчиком –**300 000.0** (триста тысяч) рублей.
Следует к перечислению**: 300 000,0** (триста тысяч) рублей.

от Исполнителя

Руководитель работы
_____ Г.Ф.Хахулин
"___" _____ 200_ г.

от Заказчика

Главный инженер
ФГУП НПО "ОРИОН"
_____ В. В. Мартьянов
"___" _____ 200_ г.

КОПИЯ ВЕРНА
ВЕДУЩИЙ ЭКОНОМИСТ
Т. В. КЛИМОВА

Исполнитель:
Московский авиационный институт (государственный технический университет)
125933, Москва А-80, ГСП-3,
Волоколамское шоссе, д.4,
ИНН 7725070752
ОФК по САО г.Москвы,
Р/с 40503810800001009002 в отделении
I Московского ГТУ Банка России, г. Москва,
БИК 044583001, л/с 06075000562.

Заказчик:
ФГУП НПО «Орион»
143045, г.Краснознаменск, МО, ул. Октябрьская, д.7, а/я 3

ИНН 5006001899
Расч/счет №40502810500000000002
в АК Промторгбанк г. Москва
Кор/счёт № 30101810800000000139.
БИК 044583139

178 от 15.05.03г.

АКТ
сдачи-приемки работ
по договору № 22560-03020 от 3.01.2002 г.

Мы, ниже подписавшиеся, представитель Исполнителя – проректор по научной работе Московского авиационного института (Государственного Технического Университета) В.Г.ВЕРЕТЕННИКОВ с одной стороны, и представитель Заказчика Генеральный директор ФГУП НПО “Орион” А.С.Демидов с другой стороны, составили настоящий акт о том, что научно-техническая продукция по договору № 22560-03020 от 3.01.2002 г. выполнена в полном объеме и в надлежащем порядке оформлена.

Краткое описание научно-технической продукции.
В процессе работы по договору:

- Осуществлена постановка задачи, определены основные требования к системе.
- Разработаны концептуальная и физическая модели базы данных, содержащей исходные показатели состояния подразделений.
- Разработаны структура и программное обеспечение для РМ командира, РМ начальника соответствующей службы и РМ администратора для поддержки управляющих решений в части боевой подготовки, службы войск и безопасности военной службы и кадрового обеспечения.
- Проведено тестирование программного обеспечения
- Осуществлена апробация ПО в конкретном подразделении по договоренности с Заказчиком.
- Составлен заключительный технический отчет, содержащий описание ПО и документацию.

Договорная цена работы по договору составляет **600 000,0** (шестьсот тысяч) рублей без НДС.
Сумма аванса, перечисленная Заказчиком –**300 000.0** (триста тысяч) рублей.
Следует к перечислению**: 300 000,0** (триста тысяч) рублей.

Работу сдал
от Исполнителя
Проректор Института
_____ В.Г.Веретенников
“__” _____ 200_ г.

М.П.

Руководитель работы
_____ Г.Ф.Хахулин
“__” _____ 200_ г.

КОПИЯ ВЕРНА
ВЕДУЩИЙ ЭКОНОМИСТ
Т.В.КЛИМОВА

Работу принял
от Заказчика
Генеральный директор
ФГУП НПО “ОРИОН”
_____ А. С. Демидов
“12” 05 2003 г.

М.П.

СОГЛАСОВАНО
Начальник 968 ВП МО РФ
_____ В.Г.Каленов
“12” 05 2003 г.

М.П.

УТВЕРЖДАЮ
ИСПОЛНИТЕЛЬ

ФГБОУ ВПО «МАИ»
Проректор
(должность)
В.А.Шевцов
(подпись, фамилия)
" ____ " ____ 2011 г.
М.П.

УТВЕРЖДАЮ
ЗАКАЗЧИК

ОАО «Российские космические системы»
заместитель генерального директора
(должность)
А.В.Чимирис
(подпись, фамилия)
" 28 " 10 2011 г.
М.П.

А К Т № 1 / 271
сдачи-приемки научно-технической продукции

Договор от " *11* " *января* 2011 г. № 41640-04020

Мы, нижеподписавшиеся, представитель Исполнителя заведующий кафедрой 402 ФГБОУ ВПО «МАИ» Мазепа Роман Богданович с одной стороны,
(должность, фамилия, имя, отчество)
и представитель Заказчика главный конструктор направления "Навигационно-временные системы" Дворкин Вячеслав Владимирович с другой стороны,
(должность, фамилия, имя, отчество)
составили настоящий акт о том, что « 28 » октября 2011 г. завершено выполнение этапа № 1 календарного плана договора

«Исследование направлений интеграции и унификации технических и программных средств, комплексов, систем в части разработки программно-аппаратного комплекса оценки качества технологий навигационно-временных определений высокодинамичных объектов и определение с его помощью точностных показателей алгоритмов оценки углов ориентации. Разработка макета системы поддержки принятия решений (СППР) на этапах развития и целевого использования глобальной навигационной системы ГЛОНАСС и реализующие его программные средства.»

Научно-техническая продукция по этапу № 1 — Архитектура программно-аппаратного комплекса оценки точностных показателей алгоритмов оценки ориентации, тестовые сценарии формирования навигационных сигналов, соответствующих движению высокодинамичных объектов, выполненные на имитаторе ИМ-2, оценки качества алгоритмов определения углов ориентации высокодинамичных объектов, макет системы поддержки принятия решений (СППР) на этапах развития и целевого использования глобальной навигационной системы ГЛОНАСС и реализующие его программные средства,

соответствуют требованиям ТЗ, условиям договора № 41640-04020 от 11.01.2011 г. надлежа-

КОПИЯ ВЕРНА
Ведущий инженер
Т. В. Климова

Продолжение на листе

к договору № 4-1640-04020
от 11.01.2011г.

щим порядком оформлены и направлены Заказчику.

(Исх. от «28»10.2011 №402-10-179).

Сведения, относящиеся к государственной тайне, в работе не использовались и в созданной НТП не содержатся

Документы, подтверждающие выполнение работ по этапу, надлежащим порядком оформлены.

Цена этапа № 1 договора 3500,0 тыс. руб. (Три миллиона пятьсот тысяч рублей), НДС не облагается

Сумма аванса, перечисленная за выполнение этапа № 1 договора 1400,0 тыс. руб. (Один миллион четыреста тысяч рублей), НДС не облагается

Следует к перечислению 2100,0 тыс.руб. (Два миллиона сто тысяч рублей), НДС не облагается

Научно-техническую продукцию сдал:

Заведующий кафедрой 402
ФГБОУ ВПО «МАИ»
(должность)

Р.Б. Мазепа
(подпись, фамилия)

"___" _____ 2011 г.

Научно-техническую продукцию принял:

(должность)

(подпись, фамилия)

"___" _____ 2011 г.

КОПИЯ ВЕРНА
ВЕДУЩИЙ ЭКОНОМИСТ
Т.В. Климова

№6/2001 от 06.02.2001г.

МИНИСТЕРСТВО ПО ДЕЛАМ ФЕДЕРАЦИИ, НАЦИОНАЛЬНОЙ И МИГРАЦИОННОЙ ПОЛИТИКИ РОССИЙСКОЙ ФЕДЕРАЦИИ

121819, Москва, ГСП-2, Трубниковский пер., 19 Телефон 203-10-88; факс 202-44-90

29.01.2001 № 14/2-661
На № _____

Закрытое акционерное общество по внедрению и использованию образцов новых информационных технологий (ЗАО «ОВИОНТ»)

125080, Факультетский пер, 10, подъезд 1.

Направляем вам Заключение на отчеты по первому этапу контракта №01/2000 от 22.06.2000 г. Программное средство «Электронная копия федерального реестра муниципальных образований Российской Федерации» и «Разработка автоматизированной системы социально-экономического мониторинга муниципальных образований»

ПЕРВЫЙ ОТЧЕТ - Программное средство **«Электронная копия федерального реестра муниципальных образований Российской Федерации»** с расширенными функциональными возможностями - состоит из 2-х частей и представляет собой комплект документов, отвечающих требованиям действующих нормативных документов (ГОСТ 19.504-79, ГОСТ 19.402-78, ГОСТ 19.301-79, ГОСТ РД-50-34.698-90 и 34.201), описывающих новую версию электронной копии реестра муниципальных образований, разработанную в соответствии с Техническим заданием вышеуказанного Контракта и призванную значительно улучшить его параметры.

Указанное программное средство разработано с применением современных технологий (CASE-технологии, база данных Microsoft SQL Server 7.0), что позволяет проводить его дальнейшую модернизацию.

Структура таблиц базы данных организована в соответствии с Инструкцией о ведении Федерального реестра муниципальных образований в

Российской Федерации, разработана на основании Положения о Федеральном реестре муниципальных образований в Российской Федерации, утвержденного Постановлением Правительства Российской Федерации от 19 января 1998 года № 65.

ЗАКЛЮЧЕНИЕ: Отчет выполнен на высоком профессиональном уровне и полностью соответствует требованиям нормативных документов.

- ВТОРОЙ ОТЧЕТ - **«Разработка автоматизированной системы социально-экономического мониторинга муниципальных образований»** подготовлен на основе ГОСТ 7.32-91 и представляет собой результаты исследования возможности совершенствования управления сферой социально-экономического развития муниципальных образований на основе внедрения систем поддержки решений (технология DSS/UTES).

Разработанная в ходе исследований автоматизированная система мониторинга является оригинальным программным продуктом, обеспечивающим не только стандартные информационные режимы (хранение, обработку, отбор и отображение информации), но и определение значений качественных показателей на основе предпочтений пользователя системы, что позволит дать оценку различных управленческих решений с точки зрения изменения состояния муниципального образования.

Программные средства данной системы разработаны на основе современных технологий (PowerBuilde, OLE-технологии, СУБД MS SQL Server 7.0) и позволяют его дальнейшую модернизацию.

ЗАКЛЮЧЕНИЕ: Отчет выполнен на высоком профессиональном уровне и полностью соответствует требованиям Технического задания и действующим нормативным документам. Вместе с тем необходимо устранить сбой нумерации разделов отчета (отсутствует раздел 4).

Исп. Трегубова Светлана Ивановна
т. 248-85-58

Первый заместитель Министра **В.А. Кирпичников**

Наименование ЗАКАЗЧИКА
Администрация г.Иваново Ивановской области
Адрес: 153001 г. Иваново, пл. Революции , 6

ИНН 3728012487
Р\С № 40206810100000310028 в ГРКЦ ЦБ РФ
г. Иваново
ОКПО 04027467, ОКОНХ 97600

Наименование ИСПОЛНИТЕЛЯ
ЗАО "ОВИОНТ"

Местонахождение: 117218, Москва, ул. Кржижановского,д.15,к.7. ИНН 7727084849

Адрес для переписки и офис для клиентов: 125080, Москва, Факультетский пер. Д. 10

Р/с № 40702810900030000005 в КБ «СДМ - Банк» по г. Москве.
К/с 30101810600000000685, БИК 044583685

АКТ

сдачи-приемки научно-технической продукции
по договору № 36/2000 от 01 марта 2000 г.
и по Дополнительному соглашению №1 от 25.12.2000г. к настоящему договору

г. Москва "25" декабря 2000 г.

Мы, нижеподписавшиеся, представитель ЗАКАЗЧИКА в лице заместителя Главы города Иваново Батова Александра Юрьевича,- с одной стороны, и представитель ИСПОЛНИТЕЛЯ в лице Генерального директора ЗАО «ОВИОНТ» Суркова Владимира Васильевича,- с другой стороны, составили настоящий АКТ о том, что НИОКР по внедрению, установке системы поддержки решений DSS/UTES и сопровождению программного продукта, включая поставку новых версий и консультации по «горячей линии» выполнены полностью и в надлежащем виде оформлены.

Стоимость работ составила 30 000 рублей 00 копеек, НДС не облагается в соответствии с инструкцией Госналогслужбы РФ от11.10.95г. № 39.

Итого: Тридцать тысяч рублей 00 копеек.

Стороны претензий друг к другу не имеют.

ЗАКАЗЧИК:

Заместитель Главы г. Иваново,
Руководитель аппарата администрации

Батов А.Ю.

ИСПОЛНИТЕЛЬ:

Генеральный директор
ЗАО "ОВИОНТ"

Сурков В.В.

К исх. от 03.11.2011 № 9422/812
К вх. от 03.11.2011 № 9406-6395

УТВЕРЖДАЮ
ИСПОЛНИТЕЛЬ
Директор ИПМ
им. М.В. Келдыша РАН

Б.Н. ЧЕТВЕРУШКИН

«03» 11 2011 г.

М.П.

УТВЕРЖДАЮ
ЗАКАЗЧИК
Заместитель генерального директора
по планированию, координации
работ, контролю исполнения и
анализу результатов НИОКР
предприятия и отрасли ФГУП
ЦНИИмаш

О.П. КЛИШЕВ

«03» 11 2011 г.

М.П.

АКТ № 1
сдачи-приемки научно-технической продукции

Договор от " 15 " августа 2011 г. № (105-1313-2011)-1313 175-2011

Шифр темы ОКР «Наука - ИПМ»

Мы, нижеподписавшиеся, представитель Исполнителя Заместитель заведующего отдела №7 ИПМ им. М.В. Келдыша РАН Осипов Владимир Петрович, с одной стороны,

и представитель Заказчика Начальник отдела ФГУП ЦНИИмаш Афанасьев Андрей Васильевич, с другой стороны,

составили настоящий акт о том, что завершено выполнение этапа № 1 календарного плана «Разработка метода и программно-математического обеспечения (ПМО) определения приоритетности исследований и экспериментов для формирования Этапной программы НПИ» (пп. 2.2.1, 2.2.2 ТЗ).
Отчетность – Технический акт, НТО, ПМО, инструкции по работе с ПМО для каждой категории пользователей

Научно-техническая продукция по этапу № 1

Технический акт №1 от 15.10.2011. НТО «Разработка метода и программно-математического обеспечения (ПМО) определения приоритетности исследований и экспериментов для формирования Этапной программы НПИ», инв. № **(105-1313-2011)-1313/175-2011-1.** ПМО, инв. № **(105-1313-2011)-1313/175-2011-1.5.** Инструкции по работе с ПМО для каждой категории пользователей: инструкция по работе с программно-математическим обеспечением для оператора, инв. № **(105-1313-2011)-1313/175-2011-1.1**, инструкция по работе с ПМО для лица принимающее решение (ЛПР), инв. № **(105-1313-2011)-1313/175-2011-1.2**, инструкция по работе с ПМО для эксперта, инв. № **(105-1313-2011)-1313/175-2011-1.3**, инструкция по работе с ПМО для администратора, инв. № **(105-1313-2011)-1313/175-2011-1.4**

Продолжение на листе 2

Акт №1 сдачи-приемки НТП по договору
от 15.08.2011 г. № (105-1313-2011)-1313/175-2011

соответствует требованиям ТЗ (пп. 2.2.1, 2.2.2) удовлетворяет условиям договора и

представлена Заказчику исх. ИПМ им. М.В. Келдыша РАН от 03.11.11 г. № 9422/812,
вх. ФГУП ЦНИИмаш от 03.11.11 г. № 9406-6395

Документы, подтверждающие выполнение работ по этапу 1 , надлежащим порядком оформлены.

Цена этапа № 1 2 000.000 тыс. руб. Два миллиона рублей *(без НДС)*

Сумма аванса, перечисленная за выполнение этапа № 1 800.000 тыс. руб.
Восемьсот тысяч рублей

Следует к перечислению 1 200.000 тыс. руб. Один миллион двести тысяч рублей *(без НДС)*

Научно-техническую продукцию сдал:

Заместитель заведующего
отделом №7 ИПМ им. М.В. Келдыша
РАН

В.П. Осипов

"02" 11 2011 г.

Научно-техническую продукцию принял:

Начальник отдела
ФГУП ЦНИИмаш

А.В. Афанасьев

"03" 11 2011 г.

Приложение к АСП №1

УТВЕРЖДАЮ
Директор ИПМ им. М.В. Келдыша РАН
_____ Б.Н. Четверушкин
_____ 2011 г.

Технический акт №1 от 15.10.2011 г.
о выполнении работ по этапу №1
договора от 15 августа 2011 г. № (105-1313-2011)-1313/175-2011 (ОКР «Наука-ИПМ»)
«Разработка метода и программно-математического обеспечения (ПМО) определения приоритетности исследований и экспериментов для формирования Этапной программы НПИ» (пп.2.2.1-2.2.2 ТЗ).

Настоящий технический акт составлен в том, что в соответствии с пунктами 2.2.1, 2.2.2 ТЗ «Разработка метода и программно-математического обеспечения определения приоритетности исследований и экспериментов для формирования Этапной программы НПИ. Выполнение работ постановщика по подготовке, реализации КЭ и обработке полученных результатов КЭ «Дальность» на борту РС МКС в 2011 и 2012 годах» и этапом № 1 календарного плана договора ОКР «Наука-ИПМ» за период с мая 2011 по октябрь 2011 года выполнены следующие работы:

1. Разработка метода определения приоритетности исследований и экспериментов для формирования Этапной программы НПИ.
2. Разработка первой части многопользовательской компьютеризированной СППР, предназначенной для ранжирования космических исследований и экспериментов на основе результатов экспертных оценок.

Выпущены:

1. НТО об ОКР «Разработка метода и программно-математического обеспечения (ПМО) определения приоритетности исследований и экспериментов для формирования Этапной программы НПИ», инв. № (105-1313-2011)-1313/175-2011-1.
2. Программно-математическое обеспечение на CD-диске, инв. № (105-1313-2011)-1313/175-2011-1.5.
3. Инструкции по работе с программно-методическим обеспечением (ПМО) для каждой категории пользователей (оператора, ЛПР (лица принимающее решение), Эксперта, Администратора), инв. №№ (105-1313-2011)-1313/175-2011-*1.1 ÷ 1.4*.

Работа по этапу №1 договора выполнена в полном объеме в соответствии с ТЗ (пп.2.2.1-2.2.2).

Руководитель договора,
ответственный исполнитель В.П. Осипов

К исх. от 16.10.2012 № 11103-9422/1706
К вх. от 18.10.2012 № 9406-7530

УТВЕРЖДАЮ
ИСПОЛНИТЕЛЬ
Директор ИПМ
им. М.В. Келдыша РАН

Б.Н. ЧЕТВЕРУШКИН

"16" октября 2012 г.

М. П.

УТВЕРЖДАЮ
ЗАКАЗЧИК
Заместитель генерального директора
по планированию, координации
работ, контролю исполнения и
анализу результатов НИОКР
предприятия и отрасли ФГУП
ЦНИИмаш

О.П. КЛИШЕВ

"16" октября 2012 г.

М. П.

АКТ № 1
сдачи-приемки научно-технической продукции

Договор от "10" мая 2012 г. № (105-1313-2011)-1313 140-2012

Шифр темы ОКР «Наука - ИПМ»

Мы, нижеподписавшиеся, представитель Исполнителя Заместитель заведующего отдела №7 ИПМ им. М.В. Келдыша РАН Осипов Владимир Петрович, с одной стороны,

и представитель Заказчика Начальник отдела ФГУП ЦНИИмаш Афанасьев Андрей Васильевич, с другой стороны,

составили настоящий акт о том, что завершено выполнение этапа № 1 календарного плана «Разработка метода и программно-математического обеспечения (ПМО) оптимизации планирования исследований и экспериментов для формирования Этапной программы НПИ» (пп. 2.2.7, 2.2.8 ТЗ, п. 12.4 доп.№1 к ТЗ).

Отчетность – Технический акт, НТО, ПМО, инструкции по работе с ПМО для каждой категории пользователей, коды ПМО

Научно-техническая продукция по этапу № 1

Технический акт №1 от 27.09.2012. НТО «Разработка метода и программно-математического обеспечения (ПМО) оптимизации планирования исследований и экспериментов для формирования Этапной программы НПИ», инв. № **(105-1313-2011)-1313/140-2012-1.** ПМО, инв. № **(105-1313-2011)-1313/140-2012-1.5.** Инструкции по работе с ПМО для каждой категории пользователей: инструкция по работе с программно-математическим обеспечением для оператора, инв. № **(105-1313-2011)-1313/140-2012-1.1**, инструкция по работе с ПМО для лица принимающее решение (ЛПР), инв. № **(105-1313-2011)-1313/140-2012-1.2**, инструкция по работе с ПМО для эксперта, инв. № **(105-1313-2011)-1313/140-2012-1.3**, инструкция по работе с ПМО для администратора, инв. № **(105-1313-2011)-1313/140-2012-1.4**, коды ПМО, инв. № **(105-1313-2011)-1313/140-2012-1.6**

Продолжение на листе 2

Акт №1 сдачи-приемки НТП по договору
от 15.08.2011 г. № (105-1313-2011)-1313/140-2012

соответствует требованиям ТЗ (пп. 2.2.7, 2.2.8) и доп.№1 к ТЗ (п.12.4) удовлетворяет условиям договора и

представлена Заказчику исх. ИПМ им. М.В. Келдыша РАН от 16.10.12 г. № 11103-9422/706,
вх. ФГУП ЦНИИмаш от 18.10.12 г. № 9406-7530

Документы, подтверждающие выполнение работ по этапу 1 , надлежащим порядком оформлены.

Цена этапа № 1 2 000.000 тыс. руб. Два миллиона рублей *(без НДС)*

Сумма аванса, перечисленная за выполнение этапа № 1 800.000 тыс. руб.
Восемьсот тысяч рублей

Следует к перечислению 1 200.000 тыс. руб. Один миллион двести тысяч рублей *(без НДС)*

Научно-техническую продукцию сдал:

Заместитель заведующего
отделом №7
ИПМ им. М.В. Келдыша РАН

В.П. Осипов

"16" октября 2012 г.

Научно-техническую продукцию принял:

Начальник отдела
ФГУП ЦНИИмаш

А.В. Афанасьев

"18" октября 2012 г.