МИНИСТЕРСТВО ОБРАЗОВАНИЯ РЕСПУБЛИКИ БЕЛАРУСЬ

БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ

ФАКУЛЬТЕТ ПРИКЛАДНОЙ МАТЕМАТИКИ И ИНФОРМАТИКИ

Кафедра информационных систем управления

ВЕРЕНИЧ ВЛАДИСЛАВ НИКОЛАЕВИЧ

**МОДЕЛИРОВАНИЕ ПРИНЯТИЯ ОПЕРАТИВНЫХ**

**РЕШЕНИЙ В КРУПНОМАСШТАБНЫХ КОМПАНИЯХ**

Курсовой проект

студента 3 курса 12 группы

| “Допустить к защите“  Зав. кафедройИСУ  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  “\_\_\_” \_\_\_\_\_\_\_\_\_\_\_ 2022 г. | **Руководитель**  Вальвачев Александр Николаевич,  к.т.н. |
| --- | --- |

Минск 2022

**СОДЕРЖАНИЕ**

[**ВВЕДЕНИЕ**](#_heading=h.z7lb05tin0uy) **4**

[**Глава 1. АНАЛИЗ ЛИТЕРАТУРЫ И ПОСТАНОВКА ЗАДАЧИ**](#_heading=h.z07mczqq7kp4) **5**

[1.1 Анализ литературы](#_heading=h.37d3ilool5kh) 5

[1.2 Постановка задачи разработки системы СППР](#_heading=h.t00pe4nugnan) 5

[1.3 Выводы](#_heading=h.ufrvbpjn2lxe) 6

[**Глава 2. МОДЕЛИ И АЛГОРИТМЫ**](#_heading=h.y85951mokkxg) **8**

[2.1 Модели принятия решений](#_heading=h.a71uex8haoza) 8

[2.2 Алгоритм синтеза оперативных решений](#_heading=h.mswhdl2ipn88) 8

[2.3 Выводы](#_heading=h.gui02y7n9pi4) 12

[**Глава 3. РАЗРАБОТКА ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ**](#_heading=h.f893bafahzz6) **13**

[3.1 Архитектура СППР](#_heading=h.2et92p0) 13

[3.2 Разработка программной системы СППР](#_heading=h.bhe5ykfc69n9) 17

[3.2.1 Программа построения модели КМК](#_heading=h.6v6zagjjbpan) 17

[3.2.2 Программа синтеза оперативных управляющих решений](#_heading=h.wrxlsiggavm6) 19

[**Глава 4. РЕШЕНИЕ ПРИКЛАДНОЙ ЗАДАЧИ**](#_heading=h.gp0gqv1zbj20) **22**

[4.1 Постановка прикладной задачи](#_heading=h.ofkqg3tyrlqt) 22

[4.2 Решение прикладной задачи на основе СППР](#_heading=h.jpjrjinem5bg) 23

[4.3 Выводы](#_heading=h.hifofxlqblap) 25

[**Заключение**](#_heading=h.tux8ggz8mh5o) **25**

[**Список использованных источников**](#_heading=h.olw88tisrmy7) **26**

[**Приложение 1**](#_heading=h.9r4grd1d9w56) **27**

**ПЕРЕЧЕНЬ УСЛОВНЫХ ОБОЗНАЧЕНИЙ, СИМВОЛОВ И ТЕРМИНОВ**

ПР - принятие решений;

ЗПР - задача принятия решения;

СППР - система поддержки принятия решений;

КМК - крупномасштабная компания

РРС - распределенная робототехнической система

СЦ - ситуационный центр

АСППР - алгоритм системы поддержки принятия решений

# ВВЕДЕНИЕ

Людям приходится принимать решения почти везде и почти всегда. В ходе военных действий, в политике, при управлении предприятием, при выборе автомобиля или варианта обмена квартиры и еще в тысячах других случаев. Занимаются люди этим интересным, нередко захватывающим и часто небезопасным делом со времен фараонов и по сей день. Поэтому достоин удивления тот факт, что люди осознали то, как они принимают решения совсем недавно (по историческим меркам) – вскоре, после Второй мировой войны. Оказалось, что схема процесса принятия решения не зависит от той области, в которой принимается решение. Иначе говоря, законы принятия решений едины для всех предметных областей.

Системы поддержки принятия решений (СППР) – компьютерные автоматизированные системы, целью которых является помощь людям, принимающим решение в сложных условиях, для полного и объективного анализа предметной деятельности. Они базируются на теории принятия решений, которая является научным направлением, включающим методологию выбора альтернатив, основанную на использовании методов многокритериальной оптимизации.

СППР возникли в результате слияния управленческих информационных систем и систем управления базами данных. В СППР используются разные методы: информационный поиск, интеллектуальный анализ данных, поиск знаний в базах данных, рассуждение на основе прецедентов, имитационное моделирование, эволюционные вычисления и генетические алгоритмы, нейронные сети, ситуационный анализ, когнитивное моделирование и др. Некоторые из этих методов были разработаны в рамках искусственного интеллекта. Близкие к СППР классы систем – это экспертные системы и автоматизированные системы управления.

# Глава 1. АНАЛИЗ ЛИТЕРАТУРЫ И ПОСТАНОВКА ЗАДАЧИ

## **1.1 Анализ литературы**

В зависимости от данных, с которыми эти системы работают, СППР условно можно разделить на оперативные, тактические и стратегические. Оперативные СППР предназначены для немедленного реагирования на изменения текущей ситуации в управлении финансово-экономическими процессами компании. Тактические СППР ориентированы на принятие решений для тактического менеджмента. Стратегические СППР ориентированы на анализ значительных объемов разнообразной информации, поступающих из различных источников для стратегического менеджмента. Важнейшей целью этих СППР является поиск наиболее рациональных вариантов развития бизнеса компании с учетом влияния различных факторов, таких как конъюнктура рынков, изменения финансовых рынков и рынков капиталов, изменения в законодательстве.

Мне необходимо разработать оперативную СППР, однако для удобства далее я буду использовать сокращение “СППР” вместо “оперативная СППР”.

## **1.2 Постановка задачи разработки системы СППР**

К наиболее актуальным типам КМК относятся крупные территориально распределенные роботизированные производства. Поэтому постановку задачи акцентируем на этой тематике.

Пусть имеется крупномасштабная компания, которая имеет **k** территориально распределенных цехов, в каждом их которых работаю **n** роботов **m** типов, каждый тип может состоять из **r** =1..j составных частей (аппаратно-программных модулей).

Роботы функционируют в условиях агрессивной среды и подвергаются негативному воздействию негативных факторов. В результате роботы получают устранимые повреждения или полностью разрушаются.

В процессе цифровизации экономики количество роботов постоянно растет и контроль их состояния и ремонта становится мировой проблемой при n🡪∞.

Требуется разработать универсальный метод оперативной оценки состояния роботов и организации их ремонта или замены (в зависимости от повреждений) и соответствующую систему имитационного моделирования.

Общую задачу целесообразно декомпозировать на ряд теоретических и практических задач, обеспечивающих в комплексе решение.  
Первая подзадача заключается в построении модели РРС, вторая – в разработке алгоритма оценки и синтеза решений по наблюдению, ремонту или замены роботов, третья - в разработке соответствующего программного обеспечения, четвертая в эмпирической проверке результатов. .

## **1.3 Выводы**

Существенную роль в практической реализации обозначенной выше проблемы играют именно информационные системы поддержки принятия решений (СППР). Процесс принятия какого-либо решения непосредственно связан с обработкой и структурированием больших объемов информации. Современные системы поддержки принятия решений, основанные на различных математических методах, позволяют заменить человеческие ресурсы на этапе обработки и структурирования исходных данных. Важно подчеркнуть роль СППР в процессе принятия решений – поддержка, принятие решения всегда остается за человеком. Однако не стоит в то же время преуменьшать роль СППР. Системы поддержки принятия решений стремительно завоевывают все большие и большие области сфер деятельности человека. Все чаще можно встретить системы, внедренные в организации с целью повышения эффективности процесса принятия решения, что обуславливает актуальность подробного изучения влияния информационных технологий на развитие систем поддержки принятия решений.

Особенно актуальна тема выбора той или иной технологии, направленной на решение определенных задач. Технологий, и, как следствие, методов, применяемых в СППР, большое множество. И выбор того или иного пути с конечной целью получить ранжирование альтернатив должен опираться на исходные данные и знания, полученные в процессе принятия решений без использования информационных технологий.

# Глава 2. МОДЕЛИ И АЛГОРИТМЫ

## **2.1 Модели принятия решений**

В данный момент времени разработан ряд моделей принятия решений в организационных системах на основе ситуационных методов управления. Все ситуационные системы организационного управления состоят из логических и вычислительных подсистем, в которых выполняется синтез оперативного решения. Тем не менее, в организационных системах управления с дополнительными внешними требованиями появляются особенности, которые не позволяют прямое использование имеющихся моделей и алгоритмов, а также организовать работу логических вычислительных подсистем. Результатом чего появляется необходимость в создании новых алгоритмов и моделей, адаптированных к условиям принятия оперативных решений для внешних требований.

## **2.2 Алгоритм синтеза оперативных решений**

Одним из методов по увеличению результативности организационных систем управления является использование в управлении альтернативных схем по синтезу решений, созданных на функционально избыточных совокупностях мер. Этот метод заключается в следующем.

*1. Синтез структурной модели функционально избыточной совокупности управляющих мер.*

Функциональная избыточность подразумевает существование управляющих мер, которые обладают одинаковой функциональностью, но различаются по составу исходных данных, времени выполнения и точности полученных результатов. Таким образом, функциональная избыточность включает вариатив (альтернативных способов) по принятию решений. В зависимости от конкретных ситуаций и внешних требований, принимаются меры, которые являются наилучшим решением, с учетом информационных и временных ограничений. В то же время предполагается, что в отличие от существующих подходов, в организационном управлении нельзя предпринимать слишком много альтернативных управляющих мер перед тем, как выбрать наиболее результативное решение. При этом первоначально необходимо проанализировать внешние требования, а далее установить состав управленческих мер, и затем определить их порядок. Такой метод уменьшает время принятия решений, но потребует разработку результативных алгоритмов для синтеза последовательности управляющих мер.

Любой вариант принятия оперативного решения определяется:

• содержанием и составом исполняемых действий;

• составом исполнителей управляющих мер;

• распределением управляющих мер по исполнителям;

• обменом информацией между управляющими мерами;

• последовательностью исполнения управляющих мер, т.е. алгоритмом синтеза оперативных решений.

Для построения функционально избыточной совокупности действий в организационной системе её необходимо описать моделью на основе ориентированного трижды взвешенного графа.

Зададимся функционально избыточной совокупностью управляющих мер в виде ориентированного трижды взвешенного графа:

T=(TV,TE,TL,TT,TK), где

TV – совокупность вершин функционально избыточной совокупности управляющих мер;

TE – совокупность дуг информационно бинарных отношений между управляющими мерами и данными;

TL – весовые значения вершин по исполнителям;

TT – весовые значения вершин длительности исполнения;

TK – весовые значения вершин, описывающих достоверность данных и обоснованность управляющих мер

*2. Общая модель адаптации синтеза решений с использованием функционально избыточной совокупности управляющих мер.*

Адаптация процесса принятия решений включает в себя поиск варианта совокупности управляющих мер, использование доступной информации, соблюдение необходимых ограничений и оптимизацию процесса принятия решений. Задачи оптимизации для сложных систем, которым могут быть назначены процессы принятия решений в организационных системах, требуют явно или неявно реализации следующих шагов:

1) синтез вариантов управляющих мер;

2) оценка показателей результативности для каждого из этих вариантов управляющих мер;

3) выбор результативного решения.

Этап синтеза заключается в формировании совокупности всех возможных вариантов выполнения аналитической работы в принятии решения, т.е. T=(TV,TE,TL,TT,TK) должна удовлетворять следующим свойствам:

• Граф T должен быть ациклическим (не иметь ориентированных циклов).

• Совокупность источников TV должны состоять из заданных до начала варианта генерирования управляющих мер данных.

• Совокупность стоков TV должны состоять из данных, представляющих результат исполнения варианта генерирования управляющих мер, т.е. принятое оперативное управленческое решение.

• Вершина, соответствующая управленческому воздействию, может быть достигнута хотя бы из одного источника. Подграф T является интегратором путей, возникающих в источнике, и заканчивается в стоке, где имеет быть результат. Это значит, что управляющее воздействие исполняется на основе известных или заданных данных.

• При реализации варианта генерирования управляющих мер в принятии решения должны отсутствовать дублирующие управляющие воздействия.

Процесс генерации управляющих мер подразделяется на три подпроцесса:

1) выделяется часть из графа T=(TV,TE,TL,TT,TK), описывающая информационные отношения между управляющими мерами – граф T1=(TV,TK) , удовлетворяющий вышеописанным свойствам. В результате осуществления этапа формируется совокупность вариантов W={w1, w2, …, ws};

2) по совокупности управляющих мер D⊂TV создается граф T2=(D,TL,TT, TK). Здесь рассчитываются методом теории расписаний временные отношения. При оптимизации временных оценок рассчитываются одновременно значения как оценок, так и совокупность временных отношений.

3) восстанавливается граф T=(TV,TE,TL,TT,TK) из частей T1 и T2, являющиеся частными структурно-параметрическими моделями. Этот этап определяется объединением совокупностей дуг соответствующих графов.

В результате реализации этапа оценивания рассчитываются следующие значения:

1) t(W)={t(w1), t(w2), …, t(ws)} – оценки времени реализации вариантов принятия оперативных решений в организационной системе;

2) G(W)={G(w1), G(w2), …, G(ws)} – значения оценивания достоверности реализации вариантов принятия оперативных решений в организационной системе.

После реализации этих двух этапов можно перейти к этапу выбора наиболее результативного варианта принятия управленческого решения в организационной системе:

• минимизация времени при принятии оперативного решения;

• выбор управленческого решения с максимальным значением достоверности.

## **2.3 Выводы**

Для решения задач оптимизации выбора варианта синтеза решений требуется спроектировать модели и алгоритмы реализации трех этапов – синтез, оценивание и выбор. Предложенная методика анализа управленческих решений в организационных системах позволяет учитывать изменчивость входных факторов, что позволяет более надежно найти оптимальное управленческое решение.

# Глава 3. РАЗРАБОТКА ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ

## **3.1 Архитектура СППР**

На сегодняшний день можно выделить четыре наиболее популярных типа архитектур СППР:

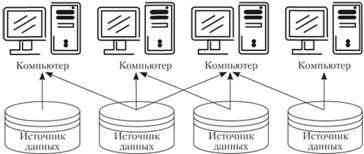
1) функциональная СППР;

2) независимые витрины данных;

3) двухуровневое хранилище данных;

4) трехуровневое хранилище данных.

**Функциональная СППР** является наиболее простой с архитектурной точки зрения (рис. 3.1). Такие системы часто встречаются на практике, в особенности в организациях с невысоким уровнем аналитической культуры и недостаточно развитой информационной инфраструктурой.



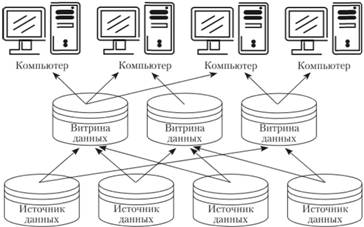
*Рис. 3.1. Функциональная CППP*

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

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

*Недостатки:* единственный источник данных, потенциально сужающий круг вопросов, на которые может ответить система; очень низкое качество данных оперативных систем с точки зрения их роли в поддержке принятия стратегических решений. В силу отсутствия этапа очистки данных данные функциональной СППР, как правило, обладают невысоким качеством; большая нагрузка на оперативную систему. Сложные запросы могут привести к остановке работы оперативной системы, что весьма нежелательно.

**Независимые витрины данных** (рис. 3.2) часто появляются в организации исторически и встречаются в крупных организациях с большим числом независимых подразделений, зачастую имеющих свои собственные отделы информационных технологий.

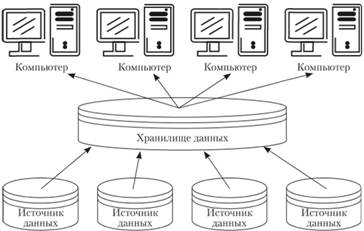


*Рис. 3.2. Независимые витрины данных*

*Преимущества:* витрины данных можно внедрять достаточно быстро; витрины проектируются для ответов на конкретный ряд вопросов; данные в витрине оптимизированы для использования определенными группами пользователей, что облегчает процедуры их наполнения, а также способствует повышению производительности.

*Недостатки*: данные хранятся многократно в различных витринах данных (это приводит к дублированию данных, а следовательно, к увеличению расходов на хранение и потенциальным проблемам, связанным с необходимостью поддержания непротиворечивости данных); процесс наполнения витрин данных при большом количестве источников данных потенциально очень сложен; данные не консолидируются на уровне предприятия, таким образом, отсутствует единая картина бизнеса.

**Двухуровневое хранилище данных** (см. рис. 3.3) строится централизованно для предоставления информации в рамках компании. Для поддержки такой архитектуры необходима выделенная команда профессионалов в области хранилищ данных. Это означает, что вся организация должна согласовать все определения и процессы преобразования данных.

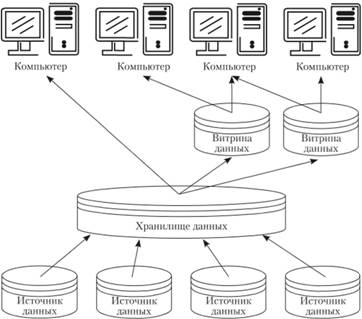


*Рис. 3.3. Двухуровневое хранилище данных*

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

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

**Трехуровневое хранилище данных** представляет собой единый централизованный источник корпоративной информации (рис. 3.4). Витрины данных представляют собой подмножества данных из хранилища, организованные для решения задач отдельных подразделений компании. Конечные пользователи имеют возможность доступа к детальным данным хранилища в случае, если данных в витрине недостаточно, а также для получения более полной картины состояния бизнеса.



*Рис. 3.4. Трехуровневое хранилище данных*

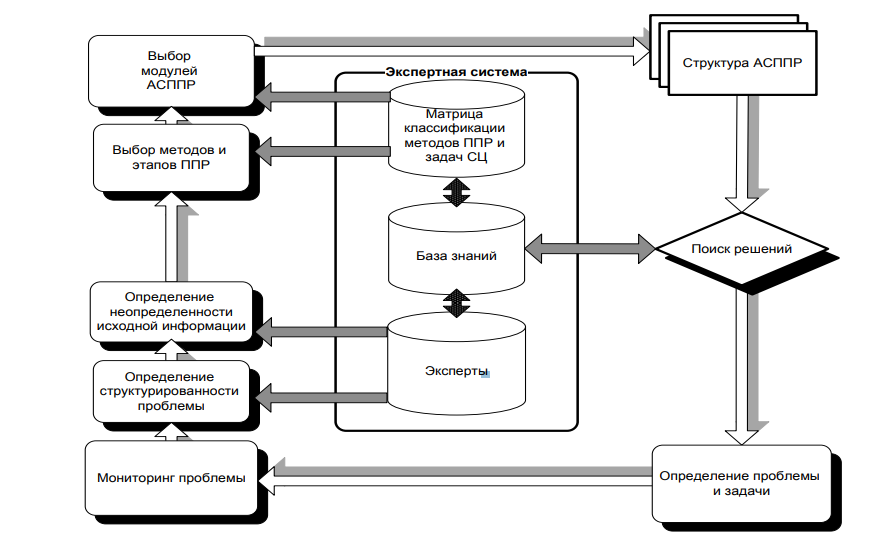
*Преимущества:* создание и наполнение витрин данных упрощено, поскольку наполнение происходит из единого стандартизованного надежного источника очищенных нормализованных данных; витрины данных синхронизированы и совместимы с корпоративным представлением (имеется корпоративная модель данных, существует возможность сравнительно легкого расширения хранилища и добавления новых витрин данных); гарантированная производительность.

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

Для курсового проекта я решил выбрать именно функциональную СППР архитектуру, так как она наиболее проста и демонстративная в данном случае.

## **3.2 Разработка программной системы СППР**

На основе мнений экспертов формируются методы принятия решений и определяются модули адаптивной СППР для разрешения возникших проблемных ситуаций. Адаптивность системы поддержки принятия решений для конкретной проблемы обеспечивается экспертной системой. На рис. 3.5 приведен алгоритм синтеза и адаптивности СППР на основе экспертной системы.



*Рис. 3.5. алгоритм синтеза и адаптивности*

*СППР на основе экспертной системы*

## **3.2.1 Программа построения модели КМК**

В построении модели участвует эксперт и программа:

# coding=utf-8

# построение модели распределенной робототехнической системы (РРС)

id = input("Введите имя РРС: \n")

#print(id)

adr = input("Введите адрес РРС: \n")

comp = input("Введите название цехов через запятую: \n")

#print(comp)

tr = input("Введите список типов роботов: \n")

sost = input("Количество модулей в каждом типе: \n")

amount = input("Введите количество роботов в каждом цеху: \n")

#print(amount)

V = input("Введите названия состояний роботов через запятую: \n")

U = input("Введите управляющие решения для состояний: \n")

alg = input("Введите имена алгоритмов оценивания для объектов компонентов: \n")

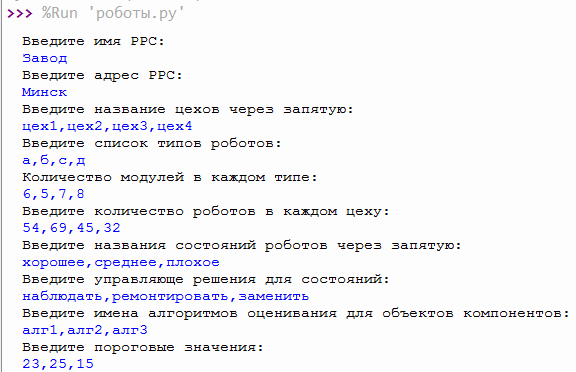
por = input("Введите пороговые значения: \n")

z = (id, adr, comp, tr, sost, amount, V, U, alg, por)

with open("c:\\DBmodel.txt", "w", encoding='utf-8') as f:

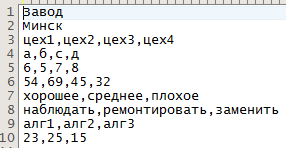
print(\*z, file = f, sep="\n")

Скрин диалога построения модели КМК:



*Рис. 3.6. Диалог построения КМК*

В результате формируется модель КМК, которую можно сохранить в txt- файле, JSON-файле или базе данных. В данной работе использован первый вариант:



*Рис. 3.7. Результат построения КМК*

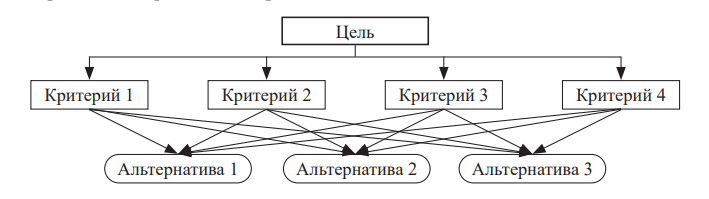
## **3.2.2 Программа синтеза оперативных управляющих решений**

В основе алгоритма программы синтеза оперативных управляющих решений лежит метод анализа иерархий (МАИ, Метод Саати) — один из наиболее распространенных методов системного подхода к разрешению сложных проблем. Существует несколько этапов для реализации данного метода.

1. Подготовительный этап, включающий в себя: определение самой проблемы, выделение альтернатив и выбор критериев и подкритериев.

2. Этап построения иерархической структуры, в которой находятся от трех до четырех уровней: цель, критерии (подкритерии) и альтернативы.

Подкритерии являются необязательным уровнем, это те же критерии, только объединенные общей характеристикой, которая при этом будет являться критерием. Итогом данного этапа является дерево критериев и альтернатив, изображенное на рис. 1.

****

*Рис. 3.7. Дерево критериев и альтернатив*

3. Третий этап в методе анализа иерархий идет в виде построения матрицы попарных сравнений критериев по цели и альтернатив по критериям.

4. Далее идет расчетный этап, где выводится итоговый вектор значений альтернатив: альтернатива X; альтернатива Y; альтернатива Z.

Каждая альтернатива имеет по набору значений, количество которых равно количеству критериев. В данном примере это три значения, для критериев A, B и C. Представим их в виде матрицы:

где

для b и c расчеты аналогичны.

Для нахождения конечного вектора значений (R) перемножаем матрицы

R = K ‧ N.

В итоге получится вектор значений вида

где — значение альтернативы X в конечном решении.

Сумма всех значений будет равняться единице, другими словами мы получили процентное значение оптимальности каждой альтернативы. Продемонстрировать работу написанной системы поддержки принятия решений и понять суть ее работы можно на примере определенной задачи. Заголовочные файлы к консольного приложения разработанного проекта расолагаются в Приложении 1.

# Глава 4. РЕШЕНИЕ ПРИКЛАДНОЙ ЗАДАЧИ

## **4.1 Постановка прикладной задачи**

На заводе имеется 3 цеха, на каждом из них запущены роботы определенного типа(a, b, c), работает определенное кол-во людей из обслуживающего персонала и располагается определенное кол-во роботов. Причем в каждом из цехов имеются роботы, находящиеся в одном из следующих состояний: хорошее, среднее, плохое. Необходимо выбрать, на какой завод отправить ремонтную бригаду, если известно, что роботы типа a более предпочтительны, чем роботы типа b, а роботы типа b более предпочтительны, чем роботы типа c. Более предпочтителен тот цех, в котором работает меньшее число сотрудников. Аналогично, чем выше доля роботов в плохом состоянии в цехе, тем более предпочтителен он для ремонтной бригады. И чем меньше доля роботов в среднем состоянии, тем более он предпочтителен для бригады. Чем больше роботов находится в цеху, тем предпочтительнее этот цех.

*Характеристика альтернатив цехов, для которых требуется принять решение*

| Критерии | Цех 1 | Цех 2 | Цех 3 |
| --- | --- | --- | --- |
| Тип роботов в цехе: | a | b | c |
| Кол-во сотрудников в цехе: | 15 | 5 | 10 |
| Кол-во роботов в цеху: | 10 | 15 | 12 |
| Кол-во роботов в среднем состоянии: | 2 | 3 | 1 |
| Кол-во роботов в плохом состоянии: | 1 | 5 | 2 |

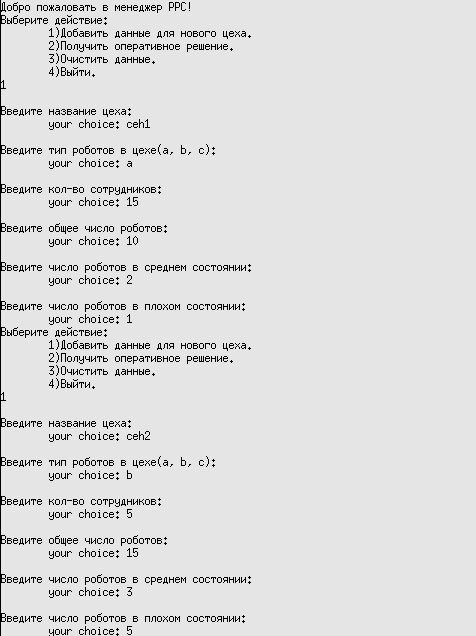
## **4.2 Решение прикладной задачи на основе СППР**

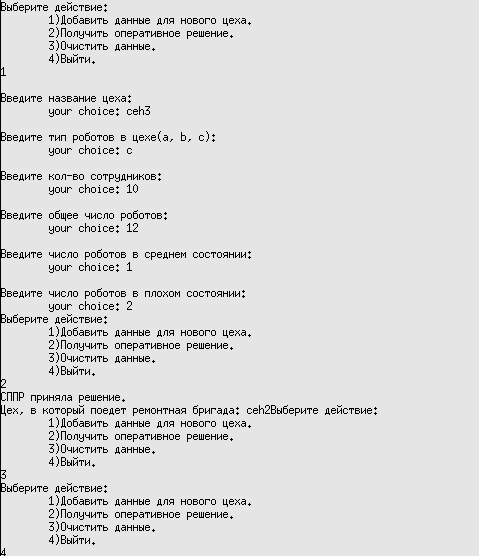
Перед вводом данных в систему необходимо внести данные об оценке критериев.

*Оценки критериев, подкритериев и альтернатив*

| Критерии | Оценка критериев | | |
| --- | --- | --- | --- |
| Тип роботов в цехе: | 2 | 1 | 0 |
| Кол-во сотрудников в цехе: | 0 | 2 | 1 |
| Кол-во роботов в каждом цеху: | 0 | 2 | 1 |
| Кол-во роботов в среднем состоянии: | 0 | 0 | 1 |
| Кол-во роботов в плохом состоянии: | 0 | 2 | 1 |
|  | 8 | 13 | 10 |

Решим поставленную задачу, используя разработтаную программную систему СППР:

****

****

## **4.3 Выводы**

Таким образом, мы получили оперативное решение, используя программную систему СППР, где была реализована архитектура функциональной СППР. В качестве алгоритма СППР использовался метод анализа иерархий.

# Заключение

Руководство различных компаний испытывает потребность в достоверной информации в оценке качества работы персонала. От этого зависит качество управления компанией, возможность эффективного планирования ее деятельности. При этом критически важными являются наглядность форм представления информации, быстрота получения новых видов отчетности, возможность анализа текущих и исторических данных. Система Поддержки Принятия Решений (СППР) предоставляют руководству такие возможности.

В данной работе были рассмотрена система поддержки принятия решений для оперативной оценки состояния роботов и организации их ремонта или замены (в зависимости от повреждений) и соответствующая система имитационного моделирования. Как уже говорилось ранее, СППР с успехом применяются в самых разных отраслях: телекоммуникациях, финансовой сфере, торговле, промышленности, медицине и многих других.

# Список использованных источников

1.Управление развитием крупномасштабных систем (MLSD’2017): труды XII-й междунар.конф,1-3 окт. 2019 г., Москва – М.:ИПУ РАН, 2019. – 1297 с.

2.Ерешко, Ф.И. Синтез организационных структур в крупномасштабных проектах цифровой экономики /Ф.И.Ерешко, Н.И.Турко, А.Д.Цвиркун, А.А.Чурсин //Автоматика и телемеханика. – 2018. – № 10. – С.121–142.

3.Виссия, Х. Принятие решений в информационном обществе / Х.Виссия,

В.В.Краснопрошин, А.Н.Вальвачев. – СПб: ЛАНЬ, 2019. – 227 с.

# Приложение 1

alternativeinterface.h:

#ifndef ALTERNATIVEINTERFACE\_H

#define ALTERNATIVEINTERFACE\_H

#include <string>

*class* AlternativeInterface {

*public*:

AlternativeInterface() = *default*;

*virtual* std::string *toString*() = 0;

};

#endif *//* *ALTERNATIVEINTERFACE\_H*

ddsdistributedroboticsystem.h:

#ifndef DDSDISTRIBUTEDROBOTICSYSTEM\_H

#define DDSDISTRIBUTEDROBOTICSYSTEM\_H

#include "decisionsupportsysteminterface.h"

#include "distributedroboticsystem.h"

*class* DDSDistributedRoboticSystem : *public* DecisionSupportSystemInterface{

*public*:

DDSDistributedRoboticSystem() = *default*;

*virtual* AlternativeInterface\* *chooseAlternative*(*const* std::vector<AlternativeInterface\*>& alternatives) *override* *final*;

std::vector<size\_t> compareAlternativesByTotalNumberOfRobots(*const* std::vector<AlternativeInterface\*>& alternatives) *const*;

std::vector<size\_t> compareAlternativesByNumberOfRobotsInMiddleState(*const* std::vector<AlternativeInterface\*>& alternatives) *const*;

std::vector<size\_t> compareAlternativesByNumberOfRobotsinBadState(*const* std::vector<AlternativeInterface\*>& alternatives) *const*;

std::vector<size\_t> compareAlternativesByNumberOfEmployees(*const* std::vector<AlternativeInterface\*>& alternatives) *const*;

std::vector<size\_t> compareAlternativesByType(*const* std::vector<AlternativeInterface\*>& alternatives) *const*;

};

#endif *//* *DDSDISTRIBUTEDROBOTICSYSTEM\_H*

decisionsupportsysteminterface.h

#ifndef DECISIONSUPPORTSYSTEM\_H

#define DECISIONSUPPORTSYSTEM\_H

#include <vector>

#include "alternativeinterface.h"

*class* DecisionSupportSystemInterface {

*public*:

DecisionSupportSystemInterface() = *default*;

*virtual* AlternativeInterface\* *chooseAlternative*(*const* std::vector<AlternativeInterface\*>& alternatives) = 0;

};

#endif *//* *DECISIONSUPPORTSYSTEM\_H*

distributedroboticsystem.h

#ifndef DISTRIBUTEDROBOTICSYSTEM\_H

#define DISTRIBUTEDROBOTICSYSTEM\_H

#include <memory>

#include <string\_view>

#include <stdio.h>

#include "typeofrobot.h"

#include "alternativeinterface.h"

*class* DistributedRoboticSystem : *public* AlternativeInterface{

*public*:

DistributedRoboticSystem();

void setNameOfRoboticSystem(*const* std::string& name);

void setTotalNumberOfRobots(*const* size\_t totalNumberOfRobots);

void setNumberOfRobotsInBadState(*const* size\_t numberOfRobotsInBadState);

void setNumberOfRobotsInMiddleState(*const* size\_t numberOfRobotsInMiddleState);

void setNumberOfEmployees(*const* size\_t numberOfEmployees);

void setTypeOfRobots(TypeOfRobot\* type);

std::string getNameOfRoboticSystem() *const*;

size\_t getTotalNumberOfRobots() *const*;

size\_t getNumberOfRobotsInBadState() *const*;

size\_t getNumberOfRobotsInMiddleState() *const*;

size\_t getNumberOfEmployees() *const*;

std::string getTypeOfRobots() *const*;

*virtual* std::string *toString*() *override* *final*;

*private*:

size\_t m\_TotalNumberOfRobots{};

size\_t m\_NumberOfRobotsInBadState{};

size\_t m\_NumberOfRobotsInMiddleState{};

size\_t m\_NumberOfEmployees{};

std::unique\_ptr<TypeOfRobot> m\_TypeOfRobot{*nullptr*};

std::string m\_NameOfRoboticSystem{};

};

#endif *//* *DISTRIBUTEDROBOTICSYSTEM\_H*

mainmenu.h:

#ifndef MAINMENU\_H

#define MAINMENU\_H

*class* MainMenu

{

*public*:

MainMenu();

void runMenu() *const*;

};

#endif *//* *MAINMENU\_H*

typeofrobot.h

#ifndef TYPEOFROBOT\_H

#define TYPEOFROBOT\_H

#include <string>

#include <string\_view>

*class* TypeOfRobot {

*public*:

TypeOfRobot() = *default*;

TypeOfRobot(*const* std::string& type);

TypeOfRobot(TypeOfRobot&& type);

TypeOfRobot(TypeOfRobot\* type);

TypeOfRobot& *operator*=(TypeOfRobot&& type);

std::string toString() *const*;

*private*:

std::string m\_Type;

};

#endif *//* *TYPEOFROBOT\_H*

**БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ**

**ФАКУЛЬТЕТ ПРИКЛАДНОЙ МАТЕМАТИКИ И ИНФОРМАТИКИ**

Кафедра информационных систем управления

“**Утверждаю**”

Заведующий кафедрой

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_В.В. Краснопрошин

“\_\_\_”   октября   2022 г.

**ЗАДАНИЕ**

ПО ПОДГОТОВКЕ КУРСОВОГО ПРОЕКТА

Студенту 3 курса Вереничу В.Н. (группа 2)

**1.Тема работы** Моделирование принятия оперативных решений в крупномасштабных компаниях

2. **Срок окончательной сдачи студентом законченной и продемонстрированной работы** 15 декабря 2022 г.

3. **Исходные данные к работе**

* Литература по принципам построения и эксплуатации КМС.
* Отчеты о деятельности КФС
* Библиотеки ПО для работы с большими данными и средствам визуализации.

Библиографические описания источников, рекомендуемых студентам к ознакомлению при выполнении работы (для изучения предметной части задания, как правило, достаточно ознакомиться с любой из перечисленных в начале списка книг):

1.Управление развитием крупномасштабных систем (MLSD’2017): труды XII-й

междунар.конф,1-3 окт. 2019 г., Москва – М.:ИПУ РАН, 2019. – 1297 с.

2.Ерешко, Ф.И. Синтез организационных структур в крупномасштабных проектах

цифровой экономики /Ф.И.Ерешко, Н.И.Турко, А.Д.Цвиркун, А.А.Чурсин //Автоматика и

телемеханика. – 2018. – № 10. – С.121–142.

3.Виссия, Х. Принятие решений в информационном обществе / Х.Виссия,

В.В.Краснопрошин, А.Н.Вальвачев. – СПб: ЛАНЬ, 2019. – 227 с.

**По желанию студента список источников может быть расширен.**

4. **Перечень вопросов подлежащих разработке или краткое содержание работы**

* Изучить назначение и структуру КМС.
* Изучить процесс и проблемы принятия решений в КМС.
* Сформулировать задачу синтеза решений.
* Построить модели и алгоритмы принятия решений.
* Разработать ПО для синтеза решений.
* На основе ПО решить прикладную задачу.

5. **Перечень графического материала** (нет)

6. **Дата выдачи задания** “\_\_\_” октября 2022 г.

7. **Календарный график** работы на весь период (с указанием этапов работы и   
сроков их выполнения)